GenRad

CHNICIAN FILE COPY

Set Preparation

ard 36535

rkstations

odel: 2275

3-D

Instruction Manual

Contents

Foreword
Introduction – Section 1
System Monitor – Section 2
File Management – Section 3
Test Set Analysis and Special Considerations – Section 4
Automatic Test Options Programming – Section 5
Circuit Description Preparation – Section 6
ATG Procedures – Section 7
Nail Assignment – Section 8
Debugging the Test Program – Section 9
Production Test Information – Section 10
Editors – Section 11
Appendices
Index

TECHNICIAN FILE COPY

Test Set Preparation

36535

227X Board Test Workstations

(Release 7.4) Model: 2275

Form 2272-0148-D

©GenRad 1983
Concord, Massachusetts, U.S.A. 01742
October, 1983

GenRad

List of Effective Pages

INSERT LATEST CHANGED PAGES. DESTROY SUPERSEDED PAGES.

NOTE: The portion of the text affected by the changes is indicated by a vertical line in the outer margin of the page.

TOTAL NUMBER OF PAGES IN THIS PUBLICATION IS 368 **CONSISTING OF THE FOLLOWING:**

Issue Original **Date** November, 1983

Page No.	Issue
Title	Original
A	Original
iii through vii	Original
1-1 through 1-33	Original
2-1 through 2-18	Original
3-1 through 3-26	Original
4-1 through 4-32	Original
5-1 through 5-15	Original
6-1 through 6-73	Original
7-1 through 7-44	Original
8-1 through 8-27	Original
9-1 through 9-26	Original
10-1 through 10-4	Original
11-1 through 11-45	Original
A-1 through A-7	Original
B-1 through B-12	Original
I-1 through I-3	Original

GenRad

INSTRUCTION MANUAL

Contents

Introduction
Differences

Orientation Notes

227X Board Test Workstation

(Release 7.4) Model: 2275

Form 2272-0155-B

©GenRad 1983
Concord, Massachusetts, U.S.A. 01742
November, 1983

GenRad

List of Effective Pages

INSERT LATEST CHANGED PAGES. DESTROY SUPERSEDED PAGES.

NOTE: The portion of the text affected by the changes is indicated by a vertical line in the outer margin of the page.

TOTAL NUMBER OF PAGES IN THIS PUBLICATION IS 8 CONSISTING OF THE FOLLOWING:

Issue Original Date November, 1983

Page No.	Issue
Title	Original
A	Original
1 through 5	Original

NOTE

The software package, described in this manual and supplied by GenRad, Inc. as a component of a system or as a separate item, is furnished for use on a single CPU. All proprietary rights are reserved by GenRad, Inc.

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in paragraph (b)(3)(B) of the Rights in Technical Data and Computer Software clause in DAR 7-104.9(a).

GenRad Inc., Concord, MA

The following are trademarks of GenRad, Inc.:

BUSBUST
GRnet
NIM
SCRATCHPROBING
SpeedPlus
TRACS

NOTE

The material in this manual is for informational purposes only and is subject to change without notice. GenRad assumes no responsibility for any errors or for consequential damages that may result from the use or misinterpretation of any of the procedures in this publication.

*The asterisk indicates pages changed, added, or deleted by the current change.

Getting Started

1.0 INTRODUCTION

Release 7.4 of the 227X Board Test Workstation software provides several new features. Some of these features present a more user-friendly interface for the test set preparation process. Another feature provides a significantly expanded testing capability for testing digital devices on 2271 and 2272 Workstations that have the optional SpeedPlus hardware. The new software also includes modifications that allow the addition of the 2275 workstation into the series of testers that use release 7.4 software.

Refer to the 227x System Manager's Guide for instructions to load the release 7.4 software onto your workstaton. Refer to the 227x Document Directory to locate subjects of interest within the manual set.

2.0 DIFFERENCES

A brief description of the new features is presented here with a reference to the manual set where it is described in more detail.

CKTGEN

Significant improvements have been made to the circuit generator (CKTGEN) which make it easier to use because of several new capabilities:

- Better prompting to lead the user through the process
- New commands provide more flexibility for user control
- Intelligence added due to new library look-up ability
- On-line circuit information readily available for display
- Messages to alert user about requirements and conflicts
- New capability to provide for user component tests
- Improved ability for modifying .CKT files.

Refer to Section 6 in the 227X Test Set Preparation manual for the specific features that contribute to these new capabilities.

SpeedPlus

The release 7.4 software supports a new optional feature, called SpeedPlus, that is available on new 2271 and 2272 Board Test Workstations. It is also available as a retrofit kit for existing 2271 and 2272 systems. When the hardware for this option is provided, testing of digital devices can be done at rates of up to 5 million test steps per second. Several features contribute to this high-speed testing capability.

- 40-MHz clock driver for precise timing ability
- 4-phase clock signals for multiple clock requirements
- Event triggering for precise drive and sense timing
- Synchronizing to on-board clocks with new ATO features
- Customizing of timing signals
- New library routines for popular microprocessor devices
- New nail assignment software to handle high-speed nails

Refer to the following manuals for information on using SpeedPlus features.

- Section 5 of 227X Test Set Preparation manual for .ATO file
- 227X Test Library Programming manual for library routines
- 227X Test Programming manual for test language enhancements

2275 Workstation

The addition of the model 2275 to the series of workstations requires a few new software descriptions in the 227X Test Set Preparation manual. These include:

Disk device descriptions in Section 3
Disk utility pages in Section 3
Nail assignment report in Section 8
Emulation of 2275 test sets in Section 8

227X Series of Workstations

The two main features of release 7.4 software are the improved CKTGEN and SpeedPlus. SpeedPlus hardware is available, as an option, on 2271 and 2272 Workstations. These models can be retrofitted to include the option. The software is the same for all users whether or not the optional hardware is installed.

Test sets generated for one 227X model can be used on other models by using the appropriate fixture converter. This is true also with the new 2275 model for the following cases:

Target Test Set	Converted Test Set
2275	2271
2275	2272
2271	2275
2270	2275

Nail assignment has been modified to include the 2275 requirements. Refer to the 227X Test Set Preparation manual for details.

User Training Board

The concept of self-training of test operators is being introduced with the 2275 Workstation. A circuit board and test fixture are provided with 2275s along with a 2275 Customer-Training Board User's Guide. This guide leads the user through the testing of a real board and explains each step and how to respond to certain board fault conditions. The training circuit is also used for examples in the 227X Test Set Preparation manual.

Digital Library Devices

There are 43 new device models added to the digital library. Seven of these are SpeedPlus models. There are 62 device models which have been updated. Eight of these have become SpeedPlus models. There are 42 new aliases added to existing models and 15 new aliases added to new models. All new and updated models and new aliases are listed in Tables 1 and 2.

Table 1
NEW LIBRARY MODELS

New Models	Description
2708	1024 X 8 ERASABLE PROGRAMMABLE READ ONLY MEMORY
2716	2048 X 8 EPROM
2732	4096 X 8 EPROM
2764	8192 X 8 EPROM
27128	16384 X 8 EPROM
555	TIMER/ONE-SHOT
74F533	TRANSPARENT OCTAL D-FLOP
74F534	FF,OCTAL,D,3-STATE
74LS640	OCTAL BUS TRANSCEIVER WITH INVERTED 3-STATE OUTPUTS
74LS641	OCTAL BUS TRANSCEIVER OPEN COLLECTOR OUTPUTS
74LS642	OCTAL BUS TRANSCEIVER INVERTED OC
74LS643	OCTAL BUS TRANSCEIVER WITH INV,NONINV 3-STATE OUTPUTS
74LS644	OCTAL BUS TRANSCEIVER INV,NONINV OPEN COLECTOR OUPUTS
74LS645	OCTAL BUS TRANSCEIVER WITH NONINVERTED 3-STATE OUTPUTS
74S350	4 BIT SHIFTER WITH 3-STATE OUTPUTS
81LS95	OCTAL BUS DRIVER
81LS96	INVERTING OCTAL BUS DRIVER
81LS97	OCTAL BUS DRIVER
81LS98	INVERTING OCTAL BUS DRIVER
8T26A	3-STATE QUAD BUS TRANSCEIVER INVERTED OUTPUT
8T28	3-STATE QUAD BUS TRANSCEIVER
9601	RETRIGGERABLE ONE-SHOT
9602	MLTVB,RETRIG,DUAL
AM25S18	QUAD D REGISTER WITH STANDARD AND THREE STATE OUTPUTS
AM26LS31	LINE DRVR, QUAD
AM26LS32	LINE RCVR, QUAD
AM9513	SYSTEM TIMING CNTLR*
AM9519	UNIVERSAL INTERRUPT CONTROLLER
DP8409	MULTI-MODE DYNAMIC RAM CONTROLLER/DRIVER*
HM50256	RAM,256K X 1*
I8031	CPU 8-BIT ROMLESS
I8051	CPU 8-BIT
I8254	PROGRAMMABLE INTERVAL TIMER*
I8279	PROGRAMMABLE KEYBOARD/DISPLAY INTERFACE
I8751	SEE I8051
IMS1420	RAM,4K X 4,DYNAMIC
MC3242A	MEMORY ADDRESS MULTIPLEXER
MC3441A	QUADRUPLE BUS TRANSCEIVER
MC3446A	QUADRUPLE BUS TRANSCEIVER
MC3447	OCTAL BUS TRANSCEIVER
MC6845	CNTLR,CRT
MC68451	MEMORY MANAGEMENT UNIT*
MC68681	DUART*
TMS4416	RAM,16K X 4

---

*Includes SpeedPlus version.

Table 2
UPDATED MODELS AND NEW ALIASES

Updated Models	New Aliases
25LS138	2148
25LS139	2149
25LS151	74F00
25LS161A	74F02
25LS163A	74F04
25LS164	74F08
25LS169A	74F10
25LS193	74F11
25LS373	74F20
25LS374	74F32
29611	74F64
68000*	74F138
7404	74F139
74107	74F151
74122	74F153
74189	74F157
74259	74F158
74390	74F175
74393	74F190
7473	74F191
7474	74F194
7476	74F240
7489	74F241
7495	74F243
74ALS112	74F244
74C85	74F253
74LS245	74F257
74LS266	74F258
74LS375	74F283
74LS670	74F352
75109A	74F353
AM2901A	74F379
AM2940	74LS51
CD4094	74LS533
CD4556	74LS534
HM4864*	74LS795
I8086*	74LS796
I8088*	74LS797
I8155	74LS798
I8156	8T96
I8251A*	8T97
I8253*	9334
I8271	AM25S08
I8272	AM25S09
I8288	AM25S10
I8355	DM8096
MC10013	DM8097
MC10135	IMS1421
MC10136	MB8168
MC10137	MC6880
MC10178	NE555

Table 2 (continued)

Updated Models	New Aliases
MC6800*	SC68681*
MC6802	SY6545
MC6809	SY6845
MC6809E	TMM2016
MC6840	
MC6850	
MK3880*	
MK3882	
MK3884	

---

*Includes SpeedPlus version.

GenRad

Production Test Division
300 Baker Avenue
Concord, Massachusetts 01742

USA

***Boston** • (Sales) Waltham, MA 02254
Tel: (617) 890-4900 • TWX: 710 324 0893
(Service) Concord, MA 01742
Tel: (617) 369-4400 • TWX: 710 347 1051

***Chicago** • 1156 Shure Drive, Arlington Heights, IL 60004
Tel: (312) 577-1881 • TWX: 910 291 1209

***Cincinnati** • Suite A, 2002 Ford Circle, Milford, OH 45150
Tel: (513) 831-9210 • TWX: 810 460 8323

***Colorado Springs** • 6180 Lehman Drive, Suite B207, Colorado Springs, CO 80907
Tel: (303) 593-8238

***Dallas** • 1121 Rockingham Lane, Suite 100, Richardson, TX 75080
Tel: (214) 234-3357 • TWX: 910 867 4771

***Los Angeles** • P.O. Box 19500, 17361 Armstrong Avenue,
Irvine Industrial Complex, Irvine, CA 92714
Tel: (714) 540-9830 • TWX: 910 595 1762

***New York** • 22-08 Route 208, Fair Lawn, NJ 07410
Tel: (NJ) (201) 797-8001; (NY) (212) 964-2722 • TWX: 710 988 2205

***Orlando** • 3751 Maguire Blvd., Suite 170, Orlando, Florida 32803
Tel: (305) 894-4303 • TWX: 810 850 0270

***San Francisco** • 2855 Bowers Avenue, Santa Clara, CA 95051
Tel: (408) 727-4400 • TWX: 910 338 0291

***Washington, D.C.** • 1701 Research Boulevard, Rockville, MD 20850
Tel: (301) 424-6224 • TWX: 710 828 9783

CANADA

***Toronto** • 307 Evans Avenue, Toronto, Ontario M8Z 1K2
Tel: (416) 252-3395 • TLX: 06-967624
Montreal • Tel: (514) 747-1052 • TLX: 05-826652

EUROPE

***England** • Norreys Drive, Maidenhead, Berkshire SL6 4BP England
Tel: (0628) 39181 • TLX: 851-848321
Best • Tel: (0) 49984240 • TLX: 844-59535
***Milan** • Tel: (02) 8466541 • TLX: 843-320373
***Munich** • Tel: (089) 41690 • TLX: 841-529917
***Paris** • Tel: (01) 7970739 • TLX: 842-220991
Rennes • Tel: (099) 535154
***Rome** • Tel: (06) 455839
Wiesbaden • Tel: (06121) 370057 • TLX: 841-04186489
***Zurich** • Tel: (01) 552420 • TLX: 845-53638

ASIA

Japan • Tokyo Electron Limited, Shinjuku-Nomura Bldg.,
1-26-2 Nishi-Shinjuku, Shinjuku-Ku, Tokyo 160, Japan
Tel: 343-4411 • TLX: 232-2240 Labtel J
Other countries in Asia and Pacific • Tel: 344-9245 (Tokyo)

LATIN AMERICA

Contact WALTHAM, MA 02254 USA
Cable GENRAD Waltham (MASS.)

*Service Facilities Available

Printed in USA

GenRad **INSTRUCTION MANUAL**

Contents

Introduction – Section 1
Operation – Section 2
Theory – Section 3

User's Guide

36535

2275 Customer Training Board

(Release 7.4)

Model: 2275

Form 2275-0122-C

Symbol ⚠ IEC 417 on equipment signifies that the manual contains information to prevent injury or equipment damage. Refer to Page iii and to other manuals in set.

©GenRad 1983
Concord, Massachusetts, U.S.A. 01742
November, 1983

List of Effective Pages

INSERT LATEST CHANGED PAGES. DESTROY SUPERSEDED PAGES.

NOTE: The portion of the text affected by the changes is indicated by a vertical line in the outer margin of the page.

TOTAL NUMBER OF PAGES IN THIS PUBLICATION IS 24 CONSISTING OF THE FOLLOWING:

Issue Original Date November, 1983
(Form 2275-0122-C)

Page No.	Issue
Title	Original
A	Original
iii through v	Original
1-1 through 1-4	Original
2-1 through 2-9	Original
3-1 through 3-3	Original

GenRad

WARRANTY

We warrant that this product is free from defects in material and workmanship and, when properly used, will perform in accordance with GenRad's applicable published specifications. If within 90 days after original shipment it is found not to meet this standard, it will be repaired or at the option of GenRad, replaced at no charge when returned to a GenRad service facility.

CHANGES IN THE PRODUCT NOT APPROVED BY GENRAD SHALL VOID THIS WARRANTY.

GENRAD SHALL NOT BE LIABLE FOR ANY INDIRECT, SPECIAL, OR CONSEQUENTIAL DAMAGES, EVEN IF NOTICE HAS BEEN GIVEN OF THE POSSIBILITY OF SUCH DAMAGES.

THIS WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

SERVICE POLICY

Your local GenRad office or representative will assist you in all matters relating to product maintenance, such as calibration, repair, replacement parts and service contracts.

GenRad policy is to maintain product repair capability for a period of five (5) years after original shipment and to make this capability available at the then prevailing schedule of charges.

See System Operation manual for Warnings and Cautions.

NOTE

The software package, described in this manual and supplied by GenRad, Inc. as a component of a system or as a separate item, is furnished for use on a single CPU. All proprietary rights are reserved by GenRad, Inc.

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in paragraph (b)(3)(B) of the Rights in Technical Data and Computer Software clause in DAR 7-104.9(a).

GenRad Inc., Concord, MA

The following are trademarks of GenRad, Inc.:

BUSBUST	SCRATCHPROBING
GRnet	SpeedPlus
NIM	TRACS

NOTE

The material in this manual is for informational purposes only and is subject to change without notice. GenRad assumes no responsibility for any errors or for consequential damages that may result from the use or misinterpretation of any of the procedures in this publication.

Table of Contents

Section and Paragraph Page

1 INTRODUCTION
- 1.1 Introduction 1-1
- 1.2 Preparing to Test 1-2

2 OPERATION
- 2.1 General 2-1
- 2.2 Testing Techniques 2-2
- 2.3 Good-Board Test 2-2
- 2.4 Fault Simulation Tests 2-3
  - 2.4.1 Switch 5 – Power Short Between UCC and Ground 2-3
  - 2.4.2 Switch 1 – Analog Component Out of Tolerance 2-4
  - 2.4.3 Switch 2 – Open Track, Single IC (SCRATCHPROBING) 2-4
  - 2.4.4 Switch 3 – Nail/Fixture Problem 2-5
  - 2.4.5 Switch 4 – Digital Component Short Circuit 2-6
  - 2.4.6 Switch 6 – BUSTEST Failure 2-7
  - 2.4.7 Switch 7 – Open Resistor 2-8
  - 2.4.8 Switch 8 – Analog Short Circuit 2-8

3 THEORY
- 3.1 Training Board Theory of Operation 3-1

Introduction – Section 1

1.1 INTRODUCTION

GenRad supplies a 2275-9615 Customer Training Board Kit (Figure 1) with each 2275 Board Test Workstation. The kit consists of the following items:

Customer Training Board	2275-4715
Training Adaptor (fixture)	2275-4006
Floppy Diskette	2275-0125
Customer Training Board User's Guide	2275-0122

Training encompasses many aspects including:

the hardware and its operation
the tester capabilities
the software modes of operation
how to test and interpret the results.

The training kit is useful as an aid in all of these areas. This manual, however, addresses the use of the training board as a tool to familiarize you with testing techniques - the basics of board testing and interpreting the results of various common messages and fault situations. The board consists of both digital and analog devices connected in a functional manner. A bank of switches is included in the circuitry to introduce short and open circuits to several of the components, resulting in circuit malfunction because of simulated component or fixture failure.

Operator training can begin before the user has developed a test set and fixture of his own. Before any board can be tested, a fixture must be built. The fixture interfaces the board type to be tested to the system. Additionally, the required testing programs must first be created (test set preparation). The system automatic test generation (ATG) software is a process that is instrumental in developing these final testing programs. There are also some interactive sub-processes (debugging, nail assignment, etc.) that are prerequisites to testing. These activities have already been accomplished with this training kit. The fixture is specifically wired to interface the training board to the tester. All of the software required to run the board tests is supplied on a floppy diskette. The testing activity can now be experienced before test set preparation for any production boards has been completed.

A familiarity with the basic operations, described in the System Operation manual, is assumed, especially those pertaining to:

Introduction to the Tester	Section 1
Controls and Indicators	Section 2
Startup and Operation	Section 3
Production Test Operation (especially the SCRATCHPROBING technique)	Section 4

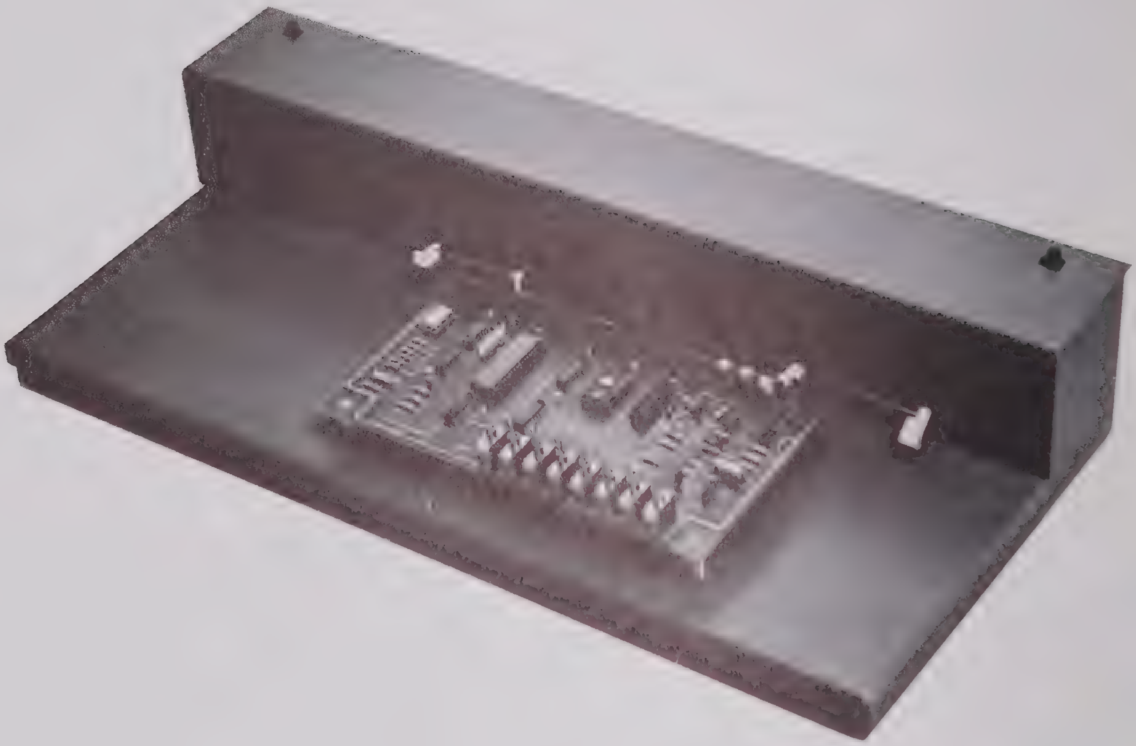


Figure 1-1. 2275 Customer Training Board and Fixture.

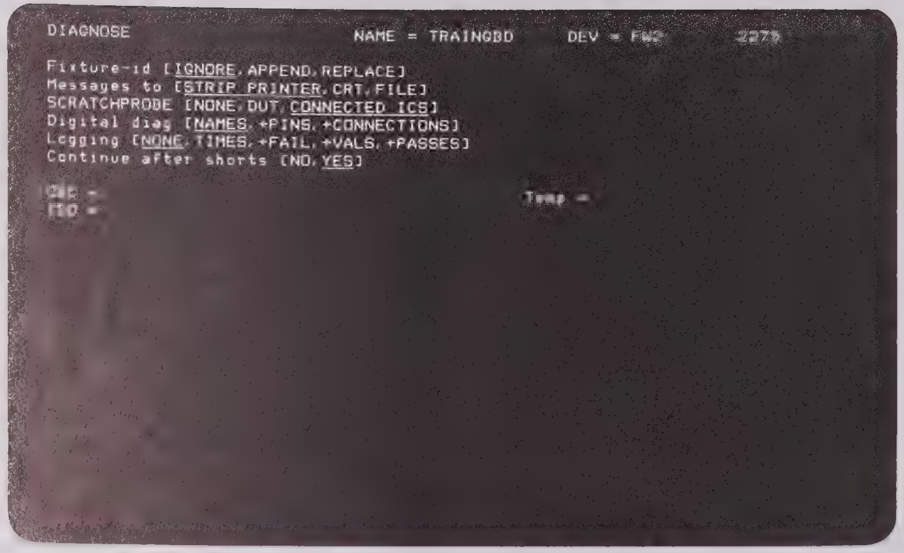
A famliarity with the procedures described in the System Operation Manual regarding basic operation is assumed, especially pertaining to:

introduction to the tester	Section 1
controls and indicators	Section 2
startup and operation	Section 3
production test operation	Section 4
(especially the SCRATCHPROBING	technique)

The GenRad training department conducts classes regarding test-set preparation. In addition, the procedures to create a test set for any board are contained in the 227X Test Set Preparation Manual. (The training board is used as an example.)

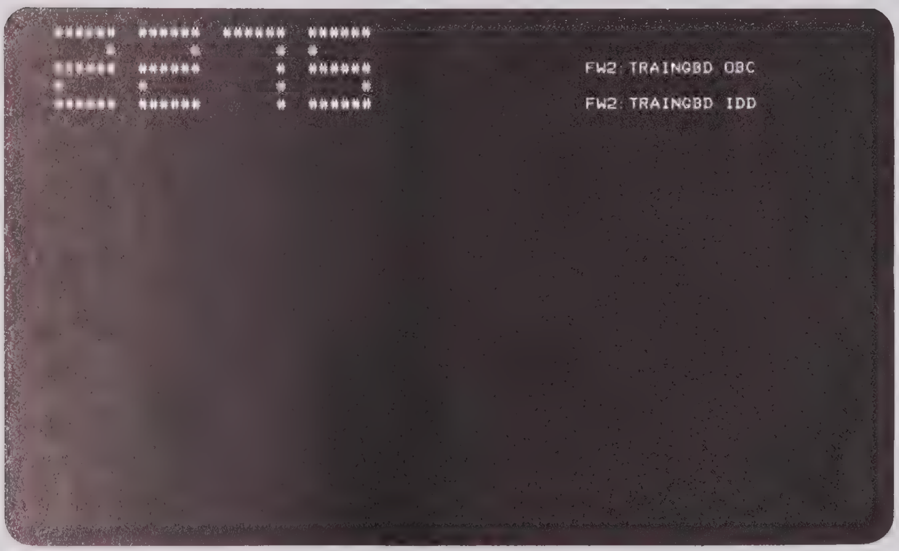
1.2 PREPARING TO TEST

Before any tests can be run, the programs TRAINGBD (.TPG,.OBC,.IDD,.SMT) must have previously been installed on the Winchester disk drive (device FW2:). If this is the initial installation, consult with the system manager to copy these files from the floppy diskette supplied to FW2: before proceeding further. Otherwise, perform the following procedure to prepare the system for training-board testing:



```
DIAGNOSE                        NAME = TRAINGBD     DEV = FW2        2275

Fixture-id [IGNORE,APPEND,REPLACE]
Messages to [STRIP PRINTER,CRT,FILE]
SCRATCHPROBE [NONE,DUT,CONNECTED ICS]
Digital diag [NAMES,+PINS,+CONNECTIONS]
Logging [NONE,TIMES,+FAIL,+VALS,+PASSES]
Continue after shorts [NO,YES]

OBC =                                          Temp =
IDD =
```

Figure 1-2. Diagnose Page Selections.



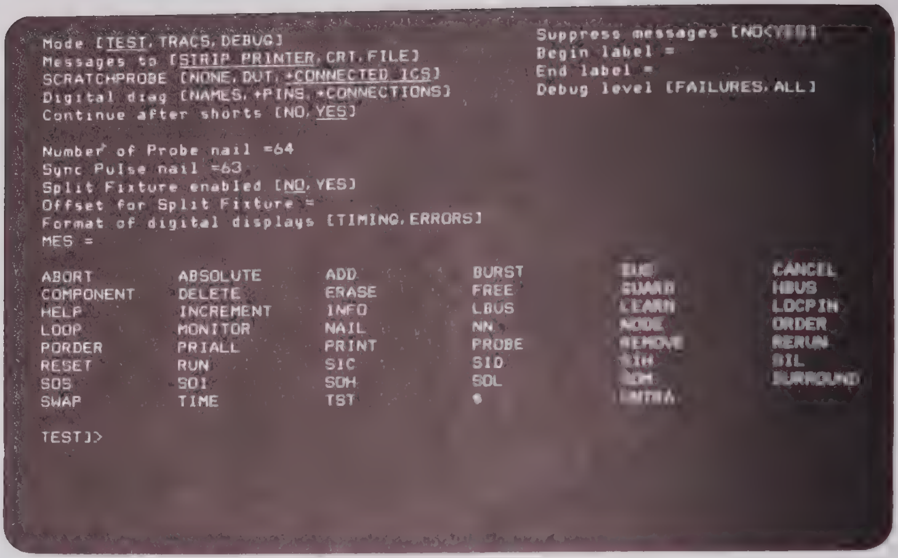
```
******  ******  ******  ******
     *       *       *  *
******  ******       *  ******               FW2 TRAINGBD OBC
*       *            *       *
******  ******       *  ******               FW2 TRAINGBD IDD
```

Figure 1-3. Test Submode Page.

f. Nails must be specified for the SYNC and SCRATCHPROBING probes. The nails are specified on the lower-level Diagnose page. Press the HELP key to display this page. Specify 63 for Sync= and 64 for Probe=; see Figure 1-4.

g. To install the training-board fixture on the system, first ensure that all tester ZIF connectors are open (turned clockwise), then gently push the fixture into position. There is an alignment-guide bar that fits into a hole near each end of the rear of the fixture. Then close each ZIF connector by turning its handle counterclockwise. The fixture is now latched to the tester.

h. To install the 2275-4715 Customer Training Board on the fixture, first ensure that the 2 ZIF connectors on the front surface of the fixture are open (turned clockwise). Next, gently push the edge connectors of the board into the sockets. Close the 2 connectors.



```
Mode [TEST,TRACS,DEBUG]                              Suppress messages [NO,YES]
Messages to [STRIP PRINTER,CRT,FILE]                 Begin label =
SCRATCHPROBE [NONE,DUT,+CONNECTED ICS]               End label =
Digital diag [NAMES,+PINS,+CONNECTIONS]              Debug level [FAILURES,ALL]
Continue after shorts [NO,YES]

Number of Probe nail =64
Sync Pulse nail =63
Split Fixture enabled [NO,YES]
Offset for Split Fixture =
Format of digital displays [TIMING,ERRORS]
MES =

ABORT        ABSOLUTE     ADD          BURST        [illegible]  CANCEL
COMPONENT    DELETE       ERASE        FREE         GUARD        HBUS
HELP         INCREMENT    INFO         LBUS         LEARN        LOCPIN
LOOP         MONITOR      NAIL         NN           NODE         ORDER
PORDER       PRIALL       PRINT        PROBE        REMOVE       RERUN
RESET        RUN          SIC          SID          SIH          SIL
SOS          SOI          SOH          SOL          SOM          SURROUND
SWAP         TIME         TST          *            [illegible]

TEST]>
```

Figure 1-4. Setting the Sync and Probe Nail Specifications on the Lower-Level Diagnose Page.

i. As an optional step, the board can be secured to the fixture. Insert 2 type 6-32 screws [approx 1 cm (3/8 in.)], one through each side of the front of the board into the the spacers provided for that purpose.

Operation – Section 2

2.1 GENERAL

The board has a bank of 8 toggle switches that are used to simulate various common faults. Figure 2-1 shows the location of these switches and the other components. Note that the left-most switch is S5, followed by S1 through S4 and S6 through S8 in that order. Some of the tests call for using the SCRATCHPROBING technique. To locate any component when the probe is used, refer to this diagram or the silk-screen labeling on the board.

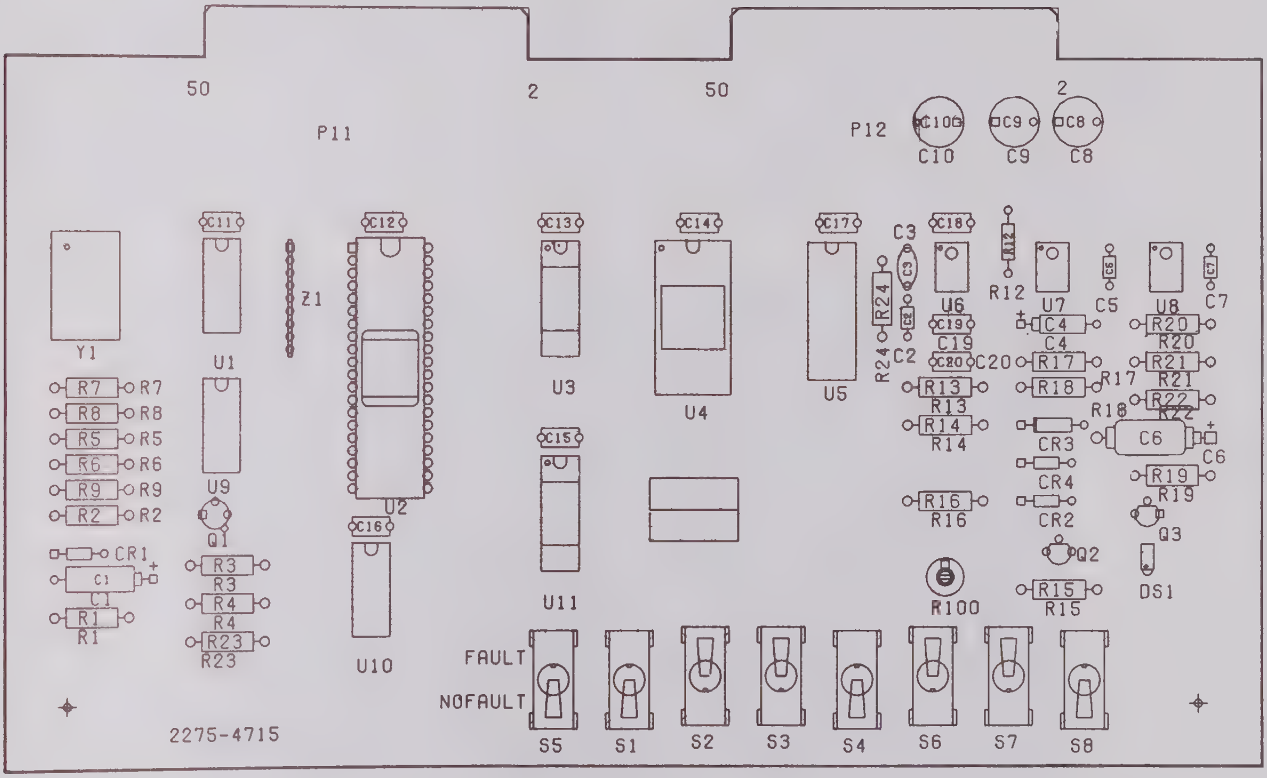

50
2
50
2
P11
P12
C10
C9
C8
C11
C12
C13
C14
C17
C18
C3
R12
Z1
Y1
U1
U6
U7
C5
U8
C7
R24
C2
C4
R20
C19
C20
R17
R21
R7
R8
R5
R6
R9
R2
U3
U4
U5
R13
R18
R22
R14
CR3
C6
U9
C15
CR4
R19
R16
CR2
Q3
Q1
U2
C16
CR1
C1
Q2
DS1
R3
R4
R23
R1
R100
R15
U11
U10
FAULT
NOFAULT
S5
S1
S2
S3
S4
S6
S7
S8
2275-4715

Figure 2-1. 2275-4715 Customer Training Board - Component Layout Drawing.

The fault-simulation tests include:

Switch 5 - Power short between VCC and GND
Switch 1 - Analog component out of tolerance
Switch 2 - Open track, single IC (SCRATCHPROBING)
Switch 3 - Nail/fixture problem
Switch 4 - Digital component short circuit
Switch 6 - BUSTEST failure
Switch 7 - Open resistor
Switch 8 - Analog short circuit

2.2 TESTING TECHNIQUES

Normally, the failure messages indict a specific component with justification. With some failures, however, the most likely source of the problem is not the indicted component. In this event, though, there is usually a clue in the error message to suggest the probable cause. This is demonstrated in a few of the tests.

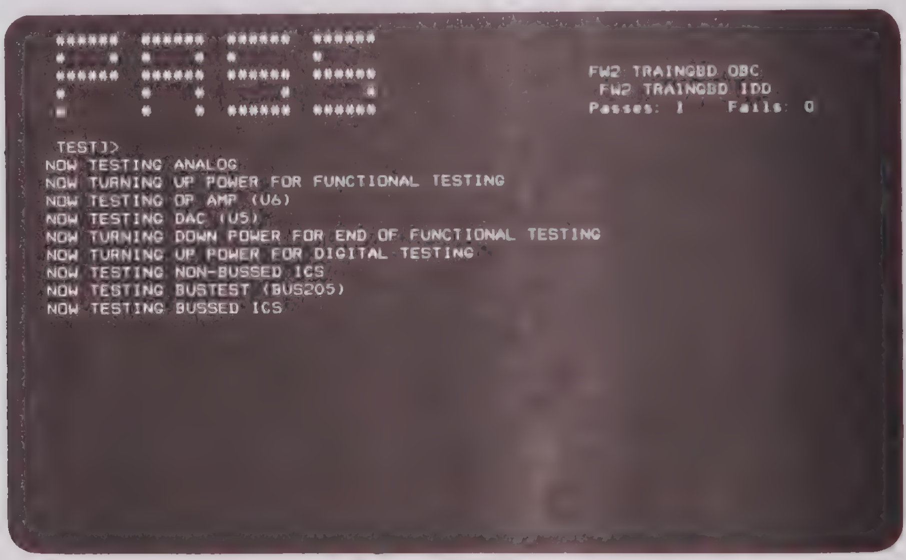
2.3 GOOD-BOARD TEST

The board is initally tested with all switches in the NOFAULT position to ensure that there are good connections between the board, the fixture and the tester. This test also ensures that the board is operational. Perform the following steps:

a. Complete the Preparing-to-Test procedure; refer to para 1.2.

b. Set the 8 switches on the training board to the NOFAULT position (flipped toward you).

c. Press <RUN>. The display shown in Figure 2-2 will appear and indicate each test as it is being run:



```
 ****** ****** ****** ******
 *    * *    * *      *
 ****** ****** ****** ******                       FW2 TRAINGBD OBC
 *      *    *      *      *                        FW2 TRAINGBD IDD
 *      *    * ****** ******                       Passes: 1     Fails: 0

 TEST]>
NOW TESTING ANALOG
NOW TURNING UP POWER FOR FUNCTIONAL TESTING
NOW TESTING OP AMP (U6)
NOW TESTING DAC (U5)
NOW TURNING DOWN POWER FOR END OF FUNCTIONAL TESTING
NOW TURNING UP POWER FOR DIGITAL TESTING
NOW TESTING NON-BUSSED ICS
NOW TESTING BUSTEST (BUS205)
NOW TESTING BUSSED ICS
```

Figure 2-2. Initial Test.

d. If the tests are successful, the PASS indication is displayed. If a single failure occurs, it is possible that a component failure has caused it. If multiple failures occur, it is likely either that the board is not seated properly or that one or more of the ZIF connectors are not fully closed. In this event, reseat the board and fixture and rerun the test. In addition, the training board has a potentiometer which might have to be readjusted to a value determined by the test program, in order for testing to start again. Perform step e, if the screen displays a prompt, requesting readjustment:

```
PREPARE TO ADJUST POT R100
(STRIKE ANY KEY TO EXIT)
RESISTANCE IS XXXXXX INCREASE TO 8.971 K OHMS
```

(where XXXXXX is the current value)

e. Using a small fine-blade screwdriver, turn the white screw-adjustment on the top of R100 (located near S6) to setting it to approximately 8.9 kilohms. (Do not force the adjustment past either stop.) The adjustment is within tolerance when the screen displays the message:

```
RESISTANCE IS XXXXXX POT IS ADJUSTED.
```

2.4 FAULT SIMULATION TESTS

2.4.1 Switch 5 - Power Short Between VCC And Ground (Figure 2-3)

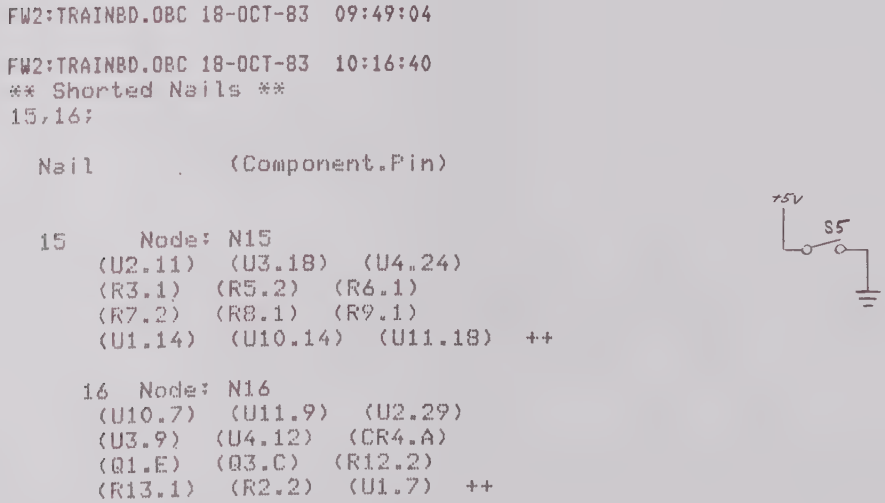
In the FAULT position, S5 introduces a short circuit between +5 V and ground. In any board testing sequence, open- and short-circuit tests are performed first. If there is a short, further testing is inhibited. This is true even if the Continue-After-Shorts option on the DIAgnose page is set to Yes, as it is in this case.

In this test the power short is readily diagnosed as follows:

a. Set S5 (left-most switch) to the FAULT position.

b. Press <RUN>. The FAIL display will appear on the screen along with the message "Power and Ground Shorted - Testing Halted", and the shorted nails message shown in Figure 2-3 is printed. The message lists the component connections on each shorted node.

c. Reset S5 to the NOFAULT position.



```
FW2:TRAINBD.OBC 18-OCT-83  09:49:04

FW2:TRAINBD.OBC 18-OCT-83  10:16:40
** Shorted Nails **
15,16;

  Nail       .    (Component.Pin)


  15       Node: N15
       (U2.11)  (U3.18)  (U4.24)
       (R3.1)  (R5.2)  (R6.1)
       (R7.2)  (R8.1)  (R9.1)
       (U1.14)  (U10.14)  (U11.18)  ++

     16  Node: N16
       (U10.7)  (U11.9)  (U2.29)
       (U3.9)  (U4.12)  (CR4.A)
       (Q1.E)  (Q3.C)  (R12.2)
       (R13.1)  (R2.2)  (U1.7)  ++
```

Figure 2-3. Switch 5 - Power-to-Ground Short Circuit.

Isolating a power short is not as easy as diagnosing it. Typically, Vcc and GND are the 2 largest nodes on a board and many components are indicted. A power capacitor is the most likely component to fail in a power-to-ground short circuit. Normally, in the production process, bare boards are tested first. For practical purposes, this insures that the problem is component-related and not a track (solder bridge) problem.

2.4.2 Switch 1 - Analog Component Out of Tolerance (Figure 2-4)

In the FAULT position, S1 closes and places two 10-kilohm resistors (R3 and R23) in parallel, causing both to measure out of tolerance. Perform the following steps:

a. Set S1 to the FAULT position.

b. Press <RUN>. The test will fail and the diagnosis indicting R3 is printed, as shown in Figure 2-4.

c. Reset S1 to the NOFAULT position.

```
R3
  EXPECTED          ACTUAL
 10K OHMS          5.024K OHMS
 LIMITS ARE 9.399K AND 10.62K OHMS

18-OCT-83  10:25:45

 BOARD 4715 FAILED
```

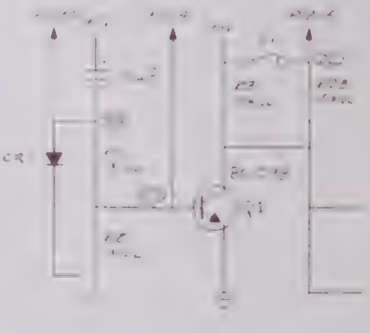


Figure 2-4. Switch 1 - Analog Component Out Of Tolerance.

2.4.3 Switch 2 - Open Track, Single IC (SCRATCHPROBING) (Figure 2-5)

In the FAULT position, S2 opens and interrupts the clock signal from the microprocessor (CPU) U2.6 (U2 pin 6). The test is run with the DIAgnose page SCRATCHPROBING option set to +CONNECTED ICS and the DIAgnose page Continue-After-Shorts option set to Yes. (These options were previously selected. in the Prepare-to-Test procedure, para 1.2.) The state of the CPU when the clock is interrupted is indeterminate. This might result in a bus failure indicated on the printout. If there is a bus failure, the display will briefly indicate that U11, U2, U3, and U4 are "untestable" because they are connected to the bus. The system then enters the SCRATCHPROBING routine. The ICs that cannot be tested because the bus failed are not listed on the printout.

Perform the following steps:

a. Set S2 to the FAULT position.

b. Press <RUN>. The FAIL display will appear on the screen and then the screen will display a SCRATCHPROBING outline of device U2. Gently probe both sides of U2. Note that U2.6 remains outside the form, even after it is reprobed.

```
17-OCT-83   10:59:30

 BOARD 4715 FAILED



U2
  Failed probing - no continuity to:(MK3880)    Z80 CPU
 Nail  272: Pin   6:
*** Track or Pin Open ***    Node: N272
```

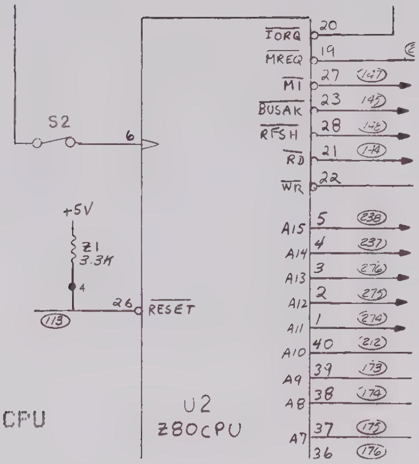

S2
6
+5V
Z1
3.3K
4
26
RESET
113
U2
Z80CPU
IORQ
20
MREQ
19
M1
27
147
BUSAK
23
145
RFSH
28
148
RD
21
174
WR
22
A15
5
238
A14
4
237
A13
3
276
A12
2
275
A11
1
274
A10
40
212
A9
39
173
A8
38
174
A7
37
175
36
176

Figure 2-5. Switch 2 - Open Track, Single IC (SCRATCHPROBING).

c. Press any key. The next prompt on the screen is a request to probe R9.2. Resistor R9 is located on the left side of the board; pin 2 is the right component lead. When this point is probed, the FAIL indication is displayed and the failure diagnosis is printed. Note that the fault has been isolated at the node between nail 272 and U2.6. (Also note that BUSTEST FAILURE does not appear on this message.)

d. Reset S2 to the NOFAULT position.

2.4.4 Switch 3 - Nail/Fixture Problem (Figure 2-6)

In the FAULT position, S3 opens. This isolates nail 111 from the node between U10.6 and U1.13. To resolve faults of this type, use the probe "+CONNECTED ICS" option, previously selected on the DIAgnose page.

```
28-SEP-83   08:44:59

 BOARD 4715 FAILED



*** No contact between track
    and nail , probable fixture problem
 Nail 111    Node: N111
       (U1.13)  (U1.10)  (U10.6)
```

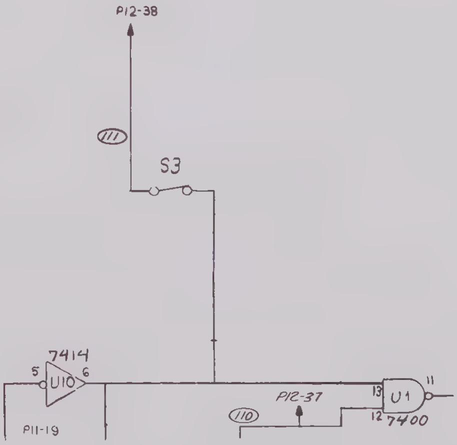

P12-38
111
S3
7414
5
U10
6
P11-19
P12-37
110
13
U1
11
12
7400

Figure 2-6. Switch 3 - Nail/Fixture Problem.

Perform the following steps:

a. Set S3 to the FAULT position.

b. Press <RUN>. The test fails.

c. A SCRATCHPROBING outline of U1, a 14-pin IC is displayed. First locate U1 on the board. Then, take the probe and check that the associated button is set to the digital position. Gently probe each pin of this IC.

d. Note after probing that U1.10 and U1.13 remain outstanding. Press any key, e.g., the Space bar. You are then prompted to probe pins U1.3, 4, 8, 12, and to reprobe U1.10. This confirms the original lack of continuity with U1.10.

e. Press any key. You are then prompted to probe U1.13. There is no continuity. Press any key.

f. A prompt to probe U10.6 (another pin on this node) is then displayed. There is no continuity. Press any key. The test aborts and the diagnosis shown in Figure 2-6 is printed.

g. Reset S3 to the NOFAULT position.

2.4.5 Switch 4 - Digital Component Short Circuit (Figure 2-7)

In the FAULT position, S4 closes and shorts the input of NAND gate U1 to its output. No IC is indicted, however, because the shorts test for the board fails before any component testing is performed. In this event, the schematic will show the relationship between nails 111 and 112, making U1 suspect. Perform the following steps:

a. Set S4 to the FAULT position.

```
** Shorted Nails **
111,112;

  Nail          (Component.Pin)


 111      Node: N111
      (U1.13)  (U1.10)  (U10.6)

    112   Node: N112
      (U1.11)  (U4.20)  (U4.18)



18-OCT-83  10:35:39

 BOARD 4715 FAILED
```

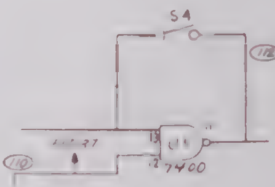


Figure 2-7. Switch 4 - Digital Component Short Circuit.

b. Press <RUN>. The screen indicates that the shorts test failed, but testing is continuing with the digital components. As long as the DIAgnose page Continue-on-shorts option is selected, other digital components (U1, U10, U11, U3) are listed as untestable. The shorts diagnosis shown in Figure 2-7 is printed.

Note that U1 has a pin connected to both nail 111 and nail 112. Further, these pins are the input and output of a gate. The inference is that the gate is the most likely fault candidate.

c. Reset S4 to the NOFAULT position.

2.4.6 Switch 6 - BUSTEST Failure (Figures 2-8 and 2-9)

In the FAULT position, S6 opens and isolates U4.18 and U4.20 This results in an EPROM data output (BUSTEST) failure. The diagnosis printed depends on whether or not probing is enabled.

Perform the following steps:

a. Set S6 to the FAULT position.

b. Press <RUN>. The test fails.

c. The display briefly indicates that the ICs associated with the bus (U11,U2,U3,U4) are untestable. The SCRATCHPROBING outline for U4 is then displayed. Using the probe, gently "scratch" all of the U4 pins.

d. Note that pins U4.18 and U4.20 remain outstanding. Probe other pins as directed. If there is no continuity, press any key to continue.

e. After U1 is probed, the display indicates the failure, and the diagnosis shown in Figure 2-8 is printed. Note from the schematic that since there is connectivity between U1.11 and nail 112, that the diagnosis is accurate.

```
18-OCT-83  10:37:09

 BOARD 4715 FAILED




U4 Failed Bustest
  Failed probing - no continuity to:(2716)
 Nail  112: Pin  18:
*** Track or Pin Open ***    Node: N112
 Nail  112: Pin  20:
*** Track'or Pin Open ***    Node: N112
```

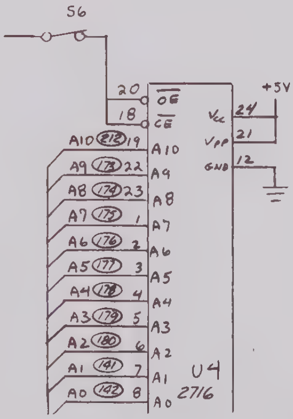


Figure 2-8. Switch 6 - BUSTEST Failure (Probing Enabled).

```
U4 Failed Bustest  (2716)

13-OCT-83  15:54:30

 BOARD 4715 FAILED
```

Figure 2-9. Switch 6 - BUSTEST Failure (Probing Disabled).

f. Disable the probe by performing an SCR=NONE <RETURN> command.

g. Press <RUN> to rerun the test. Figure 2-9 shows the diagnosis print out. The fault is not resolved beyond U4 without probing enabled. The display also indicts the bus by indicating that ICs U11, U2, U3 and U4 are untestable.

h. Reset S6 to the NOFAULT position.

2.4.7 Switch 7 - Open Resistor (Figure 2-10)

In the FAULT position, S7 opens. This disconnects the ground nail from R13, resulting in a high-impedance measurement. When there is an apparent component failure because of a very high value, an open track or component-lead/solder problem is often the cause and not the indicted component.

Perform the following steps:

a. Set S7 to the FAULT position.

b. Press <RUN>. The test fails and the diagnosis shown in Figure 2-10 is printed.

Note that the resistance measurement can vary but is typically greater than 10 megohms.

c. Reset S7 to the NOFAULT position.

```
R13
  EXPECTED          ACTUAL
 10K OHMS          20.97MEG OHMS
 LIMITS ARE 9.388K AND 10.64K OHMS

13-OCT-83   15:55:16

 BOARD 4715 FAILED
```

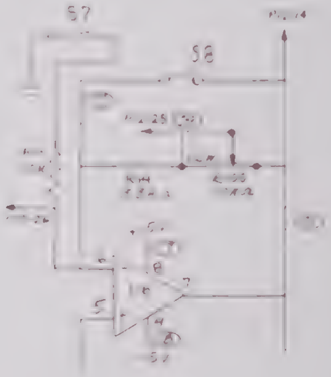


Figure 2-10. Switch 7 - Open Track Failure.

2.4.8 Switch 8 - Analog Short Circuit

In the FAULT position, S8 closes and forms a short circuit across R14/R100 as well as U6 (see Figure 2-10). Perform the following steps:

a. Set S8 to the FAULT position.

b. Press <RUN>. The test runs and a request to adjust potentiometer R100 is displayed. This component is located near S6.

c. Using a small fine-bladed screwdriver, turn the white screw-adjustment on the top of R100 approximately 1/2 turn in either direction (Do not force the adjustment past either stop.)

d. The adjustment is ineffective for reasons described below. Press any key to exit from this measurement loop.

e. The shorts test fails and the diagnosis shown in Figure 2-11 is printed.

```
R100
  EXPECTED          ACTUAL
 10K OHMS          416.1K OHMS
 LIMITS ARE 8.971K AND 11.04K OHMS
** Shorted Nails **
21,77;

  Nail          (Component.Pin)


  21      Node: N21
      (Q2.D)  (R100.3)  (R15.1)
      (U6.7)

     77  Node: N77
      (R13.2)  (R14.1)  (U6.6)



13-OCT-83  15:56:23

 BOARD 4715 FAILED
```

Figure 2-11. Switch 8 - Analog Component Short Circuit.

An interpretive look at this message shows that U6 is common to both nodes. This suggests the likelihood that either U6 is bad, that U6.6 and U6.7 are touching, or an associated solder bridge exists between the 2 tracks.

Note that there is no effect on the adjustment of R100, and that the measured value is a very high resistance. The reason for this is that the tester first sources a voltage and then measures a current to obtain a resistance value. If the measured current is in error because of a short circuit that shunts nearly all of the current, then the apparent resistance is large because the measured current is so small.

All tests have now been completed. Perform a CTRL-C command, to return to the top level of the monitor, and then an SHU command, to shut down. Then power down by pressing the red EMERGENCY POWER OFF button on the front panel of the tester.

Theory– Section 3

3.1 TRAINING BOARD THEORY OF OPERATION (Figure 3-1)

U2 is a Z80 CPU chip. This 40-pin microprocessor has associated with it a RAM memory consisting of U11 and U3. An EPROM memory (U4) contains a program which serves to drive and exercise the Z80 (U2).

The Z80 requires a clock input (U2 pin 6) of 2.5 MHz. The clock signal is derived from a 10-MHz crystal oscillator (Y1). The oscillator output is applied through 2 gates (U1) to a dual flip-flop (U9). Each section of U9 acts to divide the input frequency by 2. This produces the 2.5-MHz input to U2.

The Z80 also requires a reset input pulse. The circuitry associated with Q1 produces this conditioning signal. When power is applied to this board, Q1 turns on and a logic 0 (low) input is applied through double inversion (U10) as a low-active signal to U2. Then, as C1 located in the base circuit of Q1 charges sufficiently, Q1 turns off and the reset line to U2 goes high (inactive).

The data-bus lines (D0-D7) are applied to a D/A converter (U5). A data pattern from the Z80 is applied to U5 to produce a D/A output (Iout) of approximately 1 mA (2 mA max). This current is transformed through U6 to produce a voltage of approximately 2 V. The other section of U6 (including R14 and R100) acts as an inverting amplifier. The output of the amplifier is approximately 5 V.

Finally, there are 2 timers: U7 (1 s) and U8 (5 s) that control 2 gates, Q2 and Q3, respectively. The 5-V signal described in the previous paragraph is applied to Q2 (NJFET) which is gated by the U7 1-s timer pulse. When Q2 is turned on at 1-s intervals, the 5-V signal is fed to an LED (DS1). The cathode side of the LED is applied to ground through an SCR (Q3) which is turned on in 5-s intervals. As a result, the LED will flash approximately 8–10 times and then stay off for approximately 4 to 5 seconds.

PARTS LIST CREATED AS OF : 830801

INSTRUMENT: 2275-4715-00 ENG REV: 04 MFG REV: 01 ECO 7141 NOT ON DESC-FILE

FIND NO	LEVEL	ITEM DESCRIPTION	REF. INFO (COMP DESIG)	ITEM NO.	QTY
001	1	DUT TRAINNING BD		2275-0715-00	001
002	1	CAP Z5U .30SQ .10UF 20PCT 50V	C11-20	4400-2050-00	010
003	1	CAP Z5U TUB .01UF +80-20 100V	C2,5,7	4400-5842-00	003
004	1	CAP Z5D .310D 20PF 5PCT 500V	C3	4404-0205-00	001
005	1	CAP ALUM 10 UF 25V	C6	4450-3800-00	001
006	1	CAP TANT 1.0 UF 10PCT 35V S	C4	4450-4301-00	001
007	1	CAP TANT 10 UF 20PCT 20V	C1	4450-5100-00	001
008	1	CAP TANT 100 UF 20PCT 20V DIP S	C8-10	4450-6253-00	003
009	1	D/A CONV 8BIT UPCOMP NE5118	U5	5429-5054-00	001
010	1	ICD MK3880-4 40D MICROPROC. 4MHZ	U2	5431-2417-00	001
011	1	ICD SN7400N 14D Q 2IN PO NAND S	U1	5431-8100-00	001
012	1	ICD SN7414N 14D HX SCH TRG INV S	U10	5431-8114-00	001
013	1	ICD SN7474N 14D 2D EDTRG FL-FP S	U9	5431-8174-00	001
014	1	ICL LM1458N 8D DU OP AMP C	U6	5432-1403-00	001
015	1	ICL 555N 8DIP TIMER	U8,7	5432-1425-00	002
016	1	IC CRYSTAL OSC. 10.24MHZ .01PCTU	Y1	5434-0204-00	001
017	1	ICD SYC2114 18 1KX4 ST RAM U	U3,11	5627-1004-00	002
018	1	TRAINING PROM	U4	2275-2807-00	001
019	1	POT CERM TRM 10K OHM 20 PCT 1T	R100	6049-0109-00	001
020	1	DIODE 1N4151 75PIV IR.1UA SI	CR1,2,4	6082-1001-00	003
021	1	ZENER 1N958B 7.5V 5PCT .4W	CR3	6083-1057-00	001
022	1	LED PC (RIGHT ANGLE) 5V	DS1	6084-1108-00	001
023	1	RES COMP 330 OHM 5PCT 1/2W	R9	6100-1335-00	001
024	1	RES COMP 1.0 K 5PCT 1/2W	R1,2	6100-2105-00	002
025	1	RES COMP 3.3 K 5PCT 1/2W	R5-8,14,20	6100-2335-00	006
026	1	RES COMP 4.7 K 5PCT 1/2W	R16	6100-2475-00	001
028	1	RES COMP 10 K 5PCT 1/2W	R3,4,13,23	6100-3105-00	004
029	1	RES COMP 100 K 5PCT 1/2W	R15	6100-4105-00	001
030	1	RES COMP 390 K 5PCT 1/2W	R17	6100-4395-00	001
031	1	RES COMP 470 K 5PCT 1/2W	R22	6100-4475-00	001
032	1	RES COMP 510 K OHM 5PCT 1/2W D	R21	6100-4515-00	001
033	1	RES COMP 1.0 M 5PCT 1/2W	R18	6100-5105-00	001
034	1	RES FLM 1 OHM 1PCT 300PPM1/2WU	R19	6619-2803-00	001
035	1	RES NETWORK 8-3.3K 5PCTU	Z1	6741-0109-00	001
036	1	DESC NOT ON FILE	S1-8	7910-0850-00	008
037	1	TRANS TIS-74 F N SW PLAST	Q2	8210-1202-00	001
038	1	TRANS C203D SCR SW TO-92	Q3	8210-1215-00	001
039	1	TRANS BC-109B N S LN TO 18	Q1	8210-1248-00	001
041	1	SOCKET IC 40 CONT DIL	U2	7540-3340-00	001
042	1	SOCKET IC 24 CONTACT	U4	7540-3224-00	001
044	1	DESC NOT ON FILE	R24	6100-6105-00	001
045	1	DESC NOT ON FILE	R12	6100-2515-00	001
046	1	DESC NOT ON FILE	U5	7540-3322-00	001
		END OF PARTS LIST			

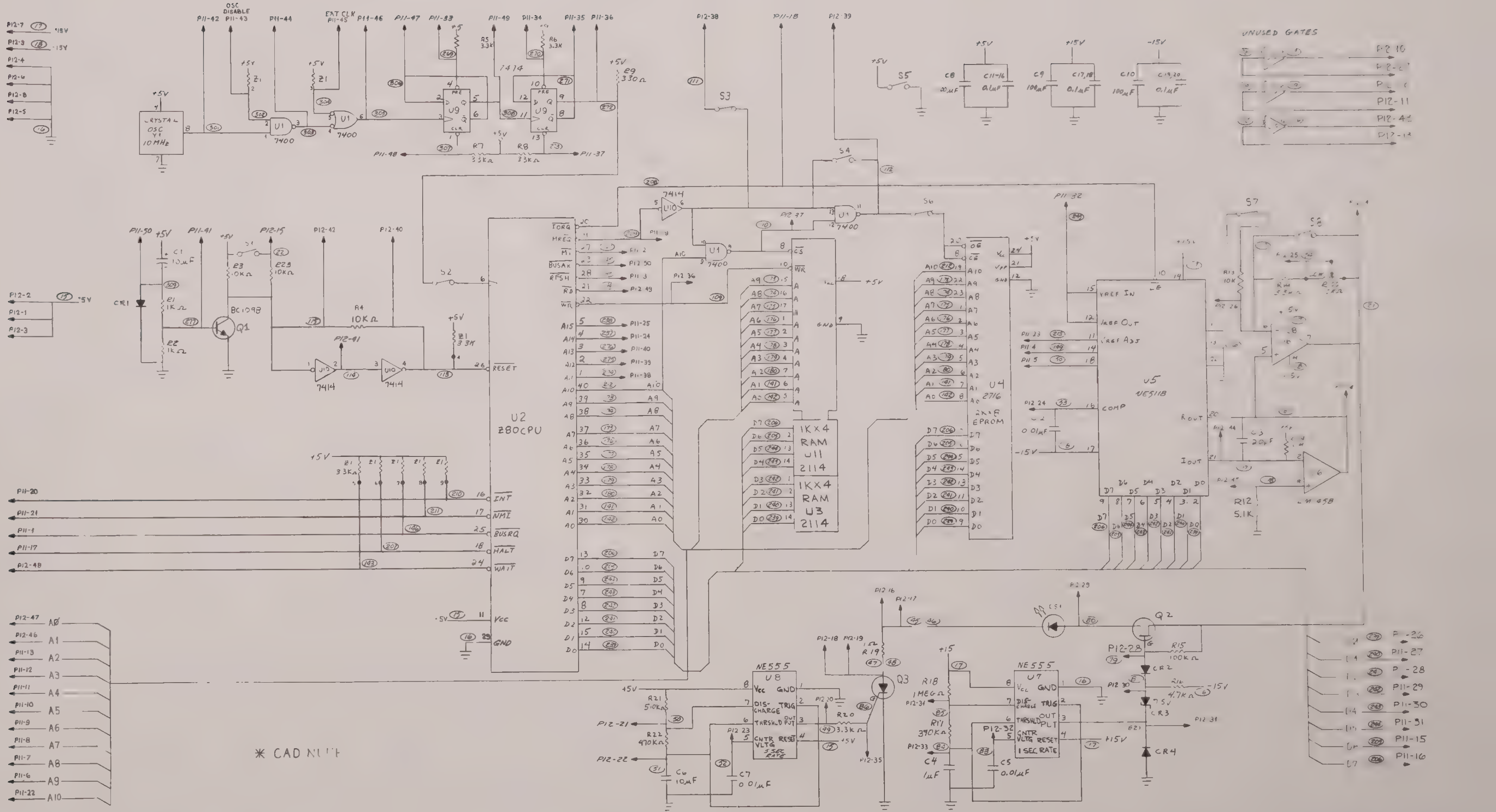

OSC DISABLE
EXT CLK
CRYSTAL OSC Y1 10MHz
U1 7400
U9
7414
R5 3.3K
R6 3.3K
R9 330Ω
R7 3.3kΩ
R8 3.3kΩ
S2
S3
S4
S5
S6
S7
S8
C8
C11-16 0.1μF
C9 100μF
C17,18 0.1μF
C10 100μF
C19,20 0.1μF
UNUSED GATES
C1 10μF
CR1
R1 1KΩ
R2 1kΩ
R3 10KΩ
R23 10KΩ
BC109B Q1
R4 10KΩ
U10 7414
Z1 3.3K
U2 Z80CPU
IORQ
MREQ
M1
BUSAK
RFSH
RD
WR
RESET
INT
NMI
BUSRQ
HALT
WAIT
Vcc
GND
1K×4 RAM U11 2114
1K×4 RAM U3 2114
U4 2716 2K×8 EPROM
U5 NE5118
VREF IN
IREF OUT
VREF ADJ
COMP
ROUT
IOUT
C3 20pF
R12 5.1K
LM 358
Q2
Q3
CS1
CR2
CR3
CR4
R15 100kΩ
R16 4.7kΩ
R18 1MEGΩ
R17 390KΩ
R19 1Ω
R20 3.3kΩ
R21 5.0kΩ
R22 470kΩ
C6 10μF
C7 0.01μF
C4 1μF
C5 0.01μF
NE555 U8
NE555 U7
DIS-CHARGE
TRIG
THRSHLD
OUT PUT
CNTR VLTG
RESET
1 SEC RATE
* CAD NUT

Figure 3-1. 2275-4715 Customer Training Board Schematic.

GenRad

Production Test Division
300 Baker Avenue
Concord, Massachusetts 01742

USA
***Boston** • (Sales) Waltham, MA 02254
Tel: (617) 890-4900 • TWX: 710 324 0893
(Service) Concord, MA 01742
Tel: (617) 369-4400 • TWX: 710 347 1051
***Chicago** • 1156 Shure Drive, Arlington Heights, IL 60004
Tel: (312) 577-1881 • TWX: 910 291 1209
***Cincinnati** • Suite A, 2002 Ford Circle, Milford, OH 45150
Tel: (513) 831-9210 • TWX: 810 460 8323
***Colorado Springs** • 6180 Lehman Drive, Suite B207, Colorado Springs, CO 80907
Tel: (303) 593-8238
***Dallas** • 1121 Rockingham Lane, Suite 100, Richardson, TX 75080
Tel: (214) 234-3357 • TWX: 910 867 4771
***Los Angeles** • P.O. Box 19500, 17361 Armstrong Avenue,
Irvine Industrial Complex, Irvine, CA 92714
Tel: (714) 540-9830 • TWX: 910 595 1762
***New York** • 22-08 Route 208, Fair Lawn, NJ 07410
Tel: (NJ) (201) 797-8001; (NY) (212) 964-2722 • TWX: 710 988 2205
***Orlando** • 3751 Maguire Blvd., Suite 170, Orlando, Florida 32803
Tel: (305) 894-4303 • TWX: 810 850 0270
***San Francisco** • 2855 Bowers Avenue, Santa Clara, CA 95051
Tel: (408) 727-4400 • TWX: 910 338 0291
***Washington, D.C.** • 1701 Research Boulevard, Rockville, MD 20850
Tel: (301) 424-6224 • TWX: 710 828 9783

CANADA
***Toronto** • 307 Evans Avenue, Toronto, Ontario M8Z 1K2
Tel: (416) 252-3395 • TLX: 06-967624
Montreal • Tel: (514) 747-1052 • TLX: 05-826652

EUROPE
***England** • Norreys Drive, Maidenhead, Berkshire SL6 4BP England
Tel (0628) 39181 • TLX 851-848321
Best • Tel (0) 49984240 • TLX 844-59535
***Milan** • Tel (02) 8466541 • TLX 843-320373
***Munich** • Tel (089) 41690 • TLX 841-529917
***Paris** • Tel (01) 7970739 • TLX 842-220991
Rennes • Tel (099) 535154
***Rome** • Tel (06) 455839
Wiesbaden • Tel (06121) 370057 • TLX 841-04186489
***Zurich** • Tel: (01) 552420 • TLX 845-53638

ASIA
Japan • Tokyo Electron Limited, Shinjuku-Nomura Bldg.,
1-26-2 Nishi-Shinjuku, Shinjuku-Ku, Tokyo 160, Japan
Tel: 343-4411 • TLX: 232-2240 Labtel J
Other countries in Asia and Pacific • Tel: 344-9245 (Tokyo)

LATIN AMERICA
Contact WALTHAM, MA 02254 USA
Cable GENRAD Waltham (MASS.)

*Service Facilities Available

Printed in USA

GenRad

INSTRUCTION MANUAL

Contents

Introduction – Section 1
System User's Index – Section 2

Document Directory

Model: 2275

227X Board Test Workstations

36535

(Release 7.4)

Form 2272-0154-C

©GenRad 1983
Concord, Massachusetts, U.S.A. 01742
November, 1983

List of Effective Pages

INSERT LATEST CHANGED PAGES. DESTROY SUPERSEDED PAGES.

NOTE: The portion of the text affected by the changes is indicated by a vertical line in the outer margin of the page.

TOTAL NUMBER OF PAGES IN THIS PUBLICATION IS 24 CONSISTING OF THE FOLLOWING:

Issue Original Date November, 1983

Page No.	Issue
Title	Original
A	Original
1-1 through 1-8	Original
2-1 through 2-14	Original

Introduction – Section 1

1.1 INTRODUCTION

This document directory describes the documentation for GenRad's 2271, 2272, and 2275 Test Systems, operating with Release 7.4 software. The directory lists all manuals supplied with each test system and briefly describes the purpose, organization and contents of each book. Also included in the directory is a cross-reference index that allows the user to locate information in any of the 227X user-type manuals.

1.2 GETTING STARTED

Delivered with each new 227X test system is a set of documentation (hardware and software) for the system. When a new software release is delivered, you also receive new manuals or change packages that describe the new software. The impact of the new software features on each manual is described on the Difference Data page near the front of that manual. Users updating to the new software release should read the difference data before referring to the manual for any operation.

1.3 MANUAL SETS

The documentation available with each 227X system is listed in Table 1-1. Some of those documents, such as the Maintenance manual, Diagrams manual, etc, describe the system hardware and are directed primarily at service personnel who must maintain the system. The rest of the documentation is oriented towards the user and covers all the facets of test set preparation and system operation. The following manuals fall in this category.

- Document Directory
- Orientation Notes
- System Operation Manual
- Test Set Preparation Manual
- Test Programming Manual
- Test Library Programming Manual
- Test Fixture Kit Instructions
- System Manager's Guide

The System Operation manual is aimed at the production test operator and contains all the information normally needed by an operator to test pc boards in a production environment.

The Orientation Notes describe the differences between Software Release 7.3 and Release 7.4.

The Test Set Preparation manual is aimed at the test engineer who must prepare the test fixture, create and debug the test program, and set up the system for the production testing of pc boards. In addition to being familiar with both the System Operation manual and Test Set Preparation manual, the test engineer should refer to the Test Programming Language manual, the Test Library Manual and the Test Fixture Kit Instructions for additional information, as needed.

The System Manager's Guide is for the individual in charge of the testing facility. It contains detailed procedures, strategies and recommendations for effectively managing a 227X system. For general information, the system manager should also be familiar with the subject matter covered in all the other user-oriented manuals.

Table 1-1
SYSTEM MANUALS (Release 7.4)

System	**Manual[1]**	**Order No.**
Common to: 2271,2272, 2275	227X Document Directory	2272-0154
	227X Orientation Notes	2272-0155
	227X Test Programming Manual	2272-0146
	227X Test Library Programming Manual	2272-0147
	227X Test Set Preparation Manual	2272-0148
	227X System Manager's Guide	2272-0153
Common to: 2271,2272	2271/2272 System Operation Manual	2272-0156
	2271/2272 Parts Lists Manual	2271-0107
	2271/2272 Site Preparation Guide	2271-0116
	Circuit Board/Module Descr., Vol. 1	2271-0150
	Circuit Board/Module Descr., Vol. 2	2271-0151
	Diagnostics Manual	2272-0166
2271	2271 Maintenance Manuals	2271-0138
	2271 Diagrams Manual	2271-0108
	2271 Service Manual[2]	2271-0149
	2271 Preventive Maintenance Card	2271-0110
	2271 Fixture Wire List Forms	2271-0118
	2271 Test Fixture Kit Assembly Instr.	2271-0113
2272	2272 Maintenance Manual	2272-0138
	2272 Diagrams Manual	2272-0108
	2272 Service Manual[2]	2272-0149
	2272 Preventive Maintenance Card	2272-0110
	2272 Test Fixture Kit Assembly Instr.	2272-0162
2275	2275 System Operation Manual	2275-0114
	2275 Customer Training Board User's Gd.	2275-0122
	2275 Site Preparation Guide	2275-0116
	2275 Test Fixture Kit Assembly Instr.	2275-0113
	2275 Maintenance Manual	2275-0104
	2275 Diagrams & Parts Lists Manual[2]	2275-0108
	2275 Unpacking Instructions	2275-0111
	2275 Service-Panel Instruction Card	2275-0112
	2275 L-Configuration Install. Instr.	2275-0132

---

[1]. Current as of Document Directory printing.
[2]. Optional manual; not supplied as part of manual set.

1.4 SYSTEM OPERATION MANUALS

System Operation manuals are available for 2271/2272 and 2275 systems. The organization and contents are essentially the same for both manuals, but each one also contains specific details applicable only to a particular system.

Section 1 - INTRODUCTION provides a general overview of the system and a brief description of both hardware and software features.

Section 2 - CONTROLS AND INDICATORS describes the functions of all operator controls and indicators.

Section 3 - PREPARING FOR SYSTEM OPERATION describes the various operating procedures that a user needs to know to operate the system (e.g., using the monitor, loading/unloading a disk cartridge, etc).

Section 4 - PRODUCTION TEST OPERATION outlines the steps involved in testing pc boards in a production test environment, including handling the test fixture and interacting with the system monitor.

Section 5 - USING THE REPORT GENERATOR describes the data logging feature by which test data can be collected and organized into concise reports.

Appendices - (A) Describes how GenRad test systems can be interconnected via a high-speed data link (GRnet(tm)) and (B) how they can be part of GenRad's Test, Repair, and Analysis Control System (TRACS(tm)).

1.5 TEST SET PREPARATION MANUAL

The organization of this manual follows the typical order in which a test program and test fixture (i.e. test set) would be prepared for board testing.

Section 1 - INTRODUCTION introduces the reader to the concepts of circuit testing with particular emphasis on in-circuit techniques.

Section 2 - SYSTEM MONITOR describes how the operator uses the monitor to interact with the system.

Section 3 - FILE MANAGEMENT describes file structures and file specifications and explains how files can be manipulated via monitor utilities.

Section 4 - TEST SET ANALYSIS AND SPECIAL CONSIDERATIONS discusses the various matters that need to be considered when starting to prepare a test set.

Section 5 - AUTOMATIC TEST OPTIONS PREPARATION shows the user how to make pre-planned changes to a test program created by ATG.

Section 6 - CIRCUIT DESCRIPTION PREPARATION describes the menu-style method of entering circuit descriptions (CKTGEN) and defines the circuit description language used in creating a .CKT file.

Section 7 - ATG PROCEDURES contains all the information needed for using the Automatic Test Generator (ATG) to compile the circuit description and create the test program.

Section 8 - NAIL ASSIGNMENT explains how the system assigns text fixture nail numbers to the test program and documents these nail assignments in various reports.

Section 9 - DEBUGGING THE TEST PROGRAM describes the system features available for translating a test program and debugging a test set.

Section 10 - PRODUCTION TEST INFORMATION shows how to help the test operator run tests in a production environment.

Section 11 - EDITORS describes how to use the system editors (TECO and ED2) to create and change files.

Appendices - Define system error messages, and list all digital devices that exist in the system library.

1.6 TEST PROGRAMMING LANGUAGE MANUAL

This manual describes the test concepts and language constructs used to code the test program. Its purpose is to provide the user with the necessary information for creating a new test program or modifying an existing one. Beginning sections (1 through 5) of the manual are aimed at introducing a new user to testing concepts, as well as to the test language. The last section (6) is a reference section containing detailed descriptions of all test language statements.

Forword - Illustrates the essential differences between 227X systems.

Section 1 - INTRODUCTION presents the purpose and scope of the manual.

Section 2 - LANGUAGE SYNTAX defines the formatting and syntactical rules pertaining to test language statements.

Section 3 - TYPES OF STATEMENTS introduces the reader to the various types of 227X test language statements.

Section 4 - TESTING THE UUT describes all the system test devices, how they are used, and how they are controlled by the test language.

Section 5 - REACTING TO A TEST FAILURE presents the various test language features available for handling test failures.

Section 6 - TEST LANGUAGE DESCRIPTIONS presents in alphabetical order for easy reference, a detailed description of every test language statement and test device in the system.

Appendices - (A) includes a glossary of terms; (B) contains an instruction summary, and (C) provides a description of external subroutines, whereby subroutines can be written in a high-level language other than the 227X test language. Appendix D describes some special uses for the SET and MEAS commands with the IEEE-488 bus.

1.7 TEST LIBRARY PROGRAMMING MANUAL

The Test Library Programming manual describes the language constructs and concepts used to code library test routines for analog, digital, and analog component libraries. It also describes procedures for creating library files using the library management monitor pages.

Section 1 - INTRODUCTION presents the 3 types of ATG libraries included in the system software.

Section 2 - AUTOMATIC TEST GENERATION PROCESS covers the ATG process and explains how the libraries interact.

Section 3 - LIBRARY MANAGEMENT USER'S GUIDE defines the procedures used to create, copy, and modify library files.

Section 4 - TEST LIBRARIAN defines the library management monitor commands.

Section 5 - ANALOG TEST LIBRARY PROGRAMMING describes the analog test selection language.

Section 6 - DIGITAL TEST LIBRARY describes the digital test structure language.

Section 7 - ANALOG COMPONENT LIBRARY USER'S GUIDE demonstrates how to write an analog component description.

Section 8 - SYSTEM LIBRARY TESTS describes the existing ATG tests and illustrates how to write new test procedures.

Appendix - MICROPROCESSOR TESTS gives specific recommendations for testing 5 popular microprocessor families.

1.8 SYSTEM MANAGER'S GUIDE

This guide contains procedures, strategies and recommendations necessary for the effective management of a 227X Test System. It is for the use of the system manager and therefore does not contain information of general use to other 227X users. The guide is organized as follows:

Section 1 - INTRODUCTION describes the system manager's role in maintaining the system.

Section 2 - SOFTWARE INSTALLATION shows how to load the operating system software into the system.

Section 3 - SYSTEM SOFTWARE MANAGEMENT describes special commands, programs and utilities that may be useful to the system manager.

Section 4 - SYSTEM ACCESS defines how system security is maintained and how user status is specified.

Section 5 - PARTITIONS AND INSTALLED TASKS discusses some issues concerning the structure of the RSX-11/M executive software.

Section 6 - DEVICE MANAGEMENT presents some device management strategies and logical device assignments.

Section 7 - TROUBLE ANALYSIS describes steps that may be taken to improve system operation and to analyze system malfunctions.

Appendix - SOFTWARE UPDATES describes how to make modifications to the system software.

1.9 HARDWARE DOCUMENTATION

A typical hardware documentation set consists of the manuals listed below. The actual manuals available with different systems may vary and are listed in Table 1-1.

.Test Fixture Kit
.Site Preparation Guide
.Maintenance Manual
.Diagrams Manual
.Parts List Manual
.Preventive Maintenance Card
.Service Manual (Optional)
.Diagnostics Manual (Optional)

1.9.1 Test Fixture Kit Assembly Instructions

These instructions provide all the necessary information for preparing the test fixture. They include the following major items:

.An introduction to the test fixture.
.Test fixturing strategies, and available system reports.
.A description of all signals available at the UUT receiver interface.
.Instructions for drilling holes in the fixture and installing the nail sockets.
.Procedures for wiring, assembling and checking out the fixture.
.Wiring and fixturing recommendations.
.Preventive maintenance procedures for the fixture.

1.9.2 Site Preparation Manual

This manual describes all the prerequisites to be considered before a 227X test system is installed or moved or before optional equipment (e.g. Foreground/Background) is added.

1.9.3 Maintenance Manual

The Maintenance manual contains an overall system-level description, maintenance procedures, system-level theory of operation, and the operation verification programs (OVPs). These OVPs are used to assure proper operation of the system modules.

1.9.4 Diagrams Manual and Parts List Manual

The theory of operation sections in the maintenance manual refer to diagrams located in the diagrams manual. The parts list manual contains item part numbers of replaceable mechanical parts and all electrical components.

1.9.5 Preventive Maintenance Card

This card specifies all preventive maintenance procedures that should be performed on the test system. This preventive maintenance schedule card should be mounted on or near the unit.

1.9.6 Service Manual

The service manual provides field service personnel with the necessary information and procedures for effective maintenance of the test systems. Included in these manuals are installation, troubleshooting, removal and replacement procedures, as well as the system OVPs.

1.9.7 Diagnostics Manual

This manual, along with the diagnostic package, helps to isolate failures and to verify the function of individual processor and test modules, as well as peripherals associated with the 227X system. The package contains detailed diagnostic and calibration procedures for the unit.

1.9.8 Circuit Board/Module Descriptions

These manuals contain the theory of operation for circuit board modules and assemblies used in test systems. The descriptions are arranged according to function (source, measurement, peripheral interface, power supplies, etc.).

1.10 ORDERING ADDITIONAL COPIES OF MANUALS

Unless otherwise noted, the documents listed in Table 1-1 for each system type are shipped as a set with each system. Additional copies of any of these manuals can be purchased directly from the Parts Sales Department, GenRad, Inc., Concord, Massachusetts, 01742, at prices effective at the time of order.

Documents covering equipment not manufactured by GenRad are furnished by the individual equipment manufacturer. Extra copies of the peripheral equipment manuals should be ordered directly from the original equipment manufacturer.

System User's Index – Section 2

The system user's index lists all the various operating and programming information required to prepare programs and to operate GenRad's circuit board test systems. Major subjects are arranged in alphabetical order, with page references to those software manuals that contain detailed information on each subject.

The system user's index should be used for locating the specific manual that contains the desired information. Readers should also refer to the detailed index in each software manual when additional information is required. Manual title abbreviations for the Users Index apply as follows:

TSPR – Test Set Preparation Manual
SOPR – System Operation Manuals
SMGD – System Manager's Guide
TLPR – Test Library Programming Manual
TPRG – Test Programming Manual

SYSTEM USERS INDEX

Item	TSPR	SOPR	SMGD	TLPR	TPRG
ACCESSORIES (Refer to Site Prep Guide)					
ACL (Analog Components Library)					
Overview				3-3	
Monitor Mode Options				4-9	
User's Guide				7-1	
ACZ (AC Impedance Meter)					
Overview					4-1
Hardware					4-4
Impedance Measurements					4-11
Test Language Statements					6-43
ANALOG TESTING					
Overview	1-1				
Measurements (general)				8-1	
Analog Instruments					4-2
Analog Channels	4-25				
Analog Test Statements					6-43
Guarding					4-9
Resistance Measurements					4-7
Kelvin Measurements					4-18
Impedance Measurements					4-11

Item	TSPR	SOPR	SMGD	TLPR	TPRG
ATG (Automatic Test Generation)					
Overview	7-1			2-1	1-1
Analog ATG	7-16			8-1	
ATG Reports	7-1				
Digital ATG	7-27			8-3	
Flow Diagram (analog)				2-7	
Monitor Mode Options	7-39				
Process Flow	7-1			2-2	
Single Component	7-43				
CDL Compiler	7-4				
Circuit Descr Extractor (CDE)	7-11			2-4	
Cross Ref Report	7-11				
Merging Programs	7-35				
Program Modifications	7-35				
ATL (Analog Test Library)					
Overview				3-2	
Monitor Mode Options				4-2	
Programming				5-1	
ATG Tests				8-1	
Resistance Test				8-9	
Potentiometer Test				8-12	
Capacitance Test				8-13	
Inductance Test				8-16	
Transistor Test				8-17	
Diode Test				8-19	
Zener Diode Test				8-21	
SCR Test				8-22	
Op Amp Test				8-24	
Integrated Circuit Test				8-28	
Analog Auto Test Gen (AATG)				2-4	
ATO (Automatic Test Options)					
Overview	5-1				
Digital Commands	5-3				
Analog Commands	5-9				
Bar-Code Wand (TRACS)		B-2			
Backdriving	1-26				4-27
Batch Mode					
Running Batch Jobs	10-2				
Creating Batch Jobs	10-2				
Options (2271/2272)		4-22			
Options (2275)		4-20			
Network-Batch Operations		A-4			
Board Test Workstations					
2271/2272		1-1			
2275		1-1			

Item	TSPR	SOPR	SMGD	TLPR	TPRG
Bus Testing					
BUSBUST	1-30				6-114
Circuit Description					
Overview	6-1				
Circuit Description Lang (CDL)	6-50				
Compiler	7-4			2-1	
Preparation	4-2				
CIRCUIT Section	6-51				
VALUE Section	6-57				
ADAPTOR Section	6-65				
Examples	6-14				
Component Types	6-54				
Flagspecs	6-70				
CKTGEN	6-1				
Nail Types	6-66				
CKTGEN					
Description	6-1				
Monitor Options	6-6				
Displays	6-9				
Help Utility	6-3				
Entering Analog Components	6-7				
Entering Digital Components	6-7				
Report Page	6-24				
Component Measurements					
Circuit Elements	6-58				
Library Routines				8-1	
Configuration Report	4-32				
Connectors					
Rear Panel (2271/2272)		2-8			
Rear Panel (2275)		2-7			
Keyboard		2-3			
Controls and Indicators		2-1			
Copy Utility	3-10				
Data Logging					
Using Data Logging		5-1			
Mode Options		5-15			
Data Collection		5-2			
Report Generator		5-14			
DCHANGE	3-11				

Item	TSPR	SOPR	SMGD	TLPR	TPRG
DCM (DC Measurement Unit)					
Hardware					4-4
Test Statements					6-47
Typical Connections					4-6
DCV (DC Voltmenter)					6-49
DCI (DC Ammeter)					6-49
DCOPY Utility	3-13				
DCS (DC Source)					
Hardware					4-5
Test Statements					6-49
Typical Connections	4-25				4-6
Optional Sources	4-25				
Debugging					
Overview	9-1				
Debug the Test Program	9-5				
Commands	9-14				
Debugging Analog Tests	9-6				4-25
Debugging Digital Tests	9-7				
Debugging Test Fixture	9-12				
Debugging SpeedPlus	9-9				
Delete Utility	3-14				
Device Names					
Overview	3-1				
2270	3-2				
2271/2272	3-4				
2275	3-5				
DFORMAT Utility	3-15				
Diagnose Mode					
Options (2271/2272)		4-16			
Options (2275)		4-14			
Limited Access		3-1			
Lower-Level Monitor	9-14				
Unlimited Access	9-12				
Diagnostic Report (2271/2272)		4-14			
Diagnostic Report (2275)		4-13			
Digital Library					
Devices	A-1				
Programming				6-1	
Process Flow				6-2	
DTSL (Digital Test Structure Lang)				6-32	
Digital Test Generator (DTG)				2-9	

Item	TSPR	SOPR	SMGD	TLPR	TPRG
Digital Testing					
Overview					6-93
Backdriving	1-25				4-27
Digital Test Statements					6-101
Driver/Sensor Circuit					4-28
Statements inserted by ATG	7-35				
VLSI Testing	1-31				4-34
Microprocessor Testing				6-20	
Directory Utility	3-17				
Disk Drive (Phoenix)					
General	3-4				
Controls		2-7			
Inserting the Disk		3-8			
Removing the Disk		3-8			
Disk Drive (Winchester)					
General	3-5				
Controls		2-6			
Inserting Floppy Disk		3-6			
Removing Floppy Disk		3-7			
Disk Drive (Hawk)					
General	3-2				
Controls		2-7			
Inserting the Disk		3-7			
Removing the Disk		3-8			
DTL (Digital Test Library)					
Overview				3-2	
Monitor Mode Options				4-5	
Editing					
Editors (General)	11-1				
Editor Utility	11-4				
Editing Commands	11-6				
Control/Escape Characters	11-10				
DEC TECO Commands	11-25				
ED2 editor	11-31				
GenRad TEX Program	11-43				
GR Terminal Function Keys	11-15				
VT10X Function Keys	11-19				
Inserting a New File	11-26				
Search and Replace	11-28				
File Recovery	11-42				
Error Messages					
Circuit Description Compiler	B-1				
Test Program Translator	B-6				
Run Time Diagnostic	B-10				
RSX-11/M System Errors	B-13				
TEX	11-43				

Item	TSPR	SOPR	SMGD	TLPR	TPRG
External Subroutines					3-7
Fail Vector					5-1
File Management					
General	3-1				
Backing-up Source Files			2-4		
Copy Utility	3-10				
Creating Files				3-3	
Dchange Utility	3-11				
Dcopy Utility	3-13				
Delete Utility	3-14				
Dformat Utility	3-15				
Directory Utility	3-17				
Editor Utility	11-4				
Free Utility	3-19				
Leditor Utility	3-19				
Managing Files	3-9				
Networking Utility		A-1			
Print Utility	3-20				
Purge Utility	3-21				
Rename Utility	3-21				
Setup Utility	4-29				
Split/Merge Utility	3-22				
Type Utility	3-24				
Unlock Utility	3-25				
System Manager			1-1		
Combining Files	3-9				
File Specifications					
General (2271/2272)	3-7	3-7			
General (2275)		3-5			
Default Extensions	3-8				
Physical/Logical Device Names	3-1				
Versions	3-7				
Wild-Card (*)	3-8				
Files, Test System	3-8				
Fixture					
Assembly (Refer to Test Fixture Manual)					
Fixture Design	4-28				
Fixture ID (2271/2272)		4-17			
Fixture ID (2275)		4-15			
Fixture Debugging	9-12				
Program Control			6-6		
Kit (2271/2272)		1-10			
Kit (2275)		1-11			
Mounting		4-3			
Split (Dual) Fixture (2271/2272)		4-6			
Split (Dual) Fixture (2275)		4-8			
Drilling	4-29				
Fixture Report (2270)	8-26				

Item	TSPR	SOPR	SMGD	TLPR	TPRG
Floppy Disk Drive					
General (2271/2272)		1-10			
General (2275)		1-5			
Controls (2271/2272)		2-8			
Controls (2275)		2-6			
Copy Files to Floppy	3-10				
Foreground/Background (See Multi-User Operation)					
Free Utility	3-19				
Guarding	1-20			8-1	4-9
GRnet					
Grounding (System) Refer to Site Preparation Guide					
Grounding (UUT)					
General					6-95
Statements	4-17				
Hardware					
Overview		1-1			4-1
Hardware Devices		1-5			4-1
System Block Diagrams					vi
Configurations	4-32				
HCS (High Current Supply)					
Hardware	4-15				
Wiring					4-51
Test Statements	4-17				6-51
Programming	4-19				
IEEE-488 Bus					
Functional Description	4-26				4-49
Test Language Statements					6-52
Wiring	4-26				
Installation (Refer to Site Preparation Guide)					
Keyboard/Keypad					
Use (2271/2272)		4-7			
Use (2275)		4-5			
Keyboard (2271/2272)		2-1			
Keyboard (2275)		4-5			
Keypad (2271/2272)		2-2			
Keypad (2275)		2-3			
TECO 227X Keypad Uses	11-17				
ED2 VT10X Keypad Use	11-20				
VT10X Gold Key	11-34				
Keywords (Test Language)					2-2

Item	TSPR	SOPR	SMGD	TLPR	TPRG
LEDITOR Utility	3-19				
Library					
Overview				1-1	
Analog Test Library Prog				5-1	
Analog Test Selection Lang (ATSL)				5-24	
Digital Library Devices	A-1				
Digital Library Programming				6-1	
Digital Test Structure Lang (DTSL)				6-51	
Test Librarian				4-1	
Library Management				3-1	
Test Format Instructions				5-41	
Digital Test Generator (DTG)				2-9	
Merge Task				2-9	
Logging Report Generator		5-8			
Microprocessor Testing				A-1	
Modifying the Test Program					
ATG Options	7-39				
ATG Modifications	7-35				
Debug Procedures	9-1				
Modify Mode Options	9-25				
Update Mode Options	9-24				
Monitor					
General	2-1				
Major Modes and Utilities	2-2				
Operation	2-6	3-1			
Screen Editing	2-14				
Single-line Commands	2-12				
Top-level Commands	2-7				
Lower Level (2271/2272)		4-9			
Lower Level (2275)		4-7			
Help Facilities	2-17				
Monitor Modes/Utilities					
General	2-2				
ACL				4-9	
ATG	7-1				
ATL				4-1	
BATCH (2271/2272)		4-21			
BATCH (2275)		4-20			
CKTGEN	6-1				
COPY	3-10				
DCHANGE	3-11				
DCOPY	3-13				
DEBUG (Diagnose)	9-5				
DELETE	3-14				
DFORMAT	3-15				
DIAGNOSE, Limited (2271/2272)		4-8			

Item	TSPR	SOPR	SMGD	TLPR	TPRG
Monitor Modes (Cont'd)					
DIAGNOSE, Limited (2275)		4-6			
DIAGNOSE, Unlimited (2271/2272)	9-12	4-17			
DIAGNOSE, Unlimited (2275)	9-12	4-15			
DIRECTORY	3-17				
DTL				4-5	
EDITOR	11-4				
FIXTURE REPORTS	8-20				
FREE	3-19				
LEDITOR	3-19				
LOG		5-15			
MODIFY	9-25				
NAIL ASSIGNMENT	8-1				
NETWORK		A-1			
PRINT	3-20				
PURGE	3-21				
RENAME	3-21				
SETUP	4-29				
SPLIT/MERGE	3-22				
TEST, Diagnose (2271/2272)		4-16			
TEST, Diagnose (2275)		4-14			
TRACS (Diagnose)		B-3			
TRANSLATE	9-1				
TYPE	3-24				
UNLOCK	3-25				
UPDATE	9-24				
Multi-User Operation					
General (2271/2272)		1-12			
General (2275)		1-11			
Considerations (2271/2272)		3-10			
Considerations (2275)		3-7			
Effective Operation			6-3		
Installation (Refer to Site Prep Guide)					
Log-on			4-1		
Protection			3-1		
MUX					
Differences between Systems					vi
Hardware					6-64
Test Statements					6-62
Nail Assignment					
Overview	8-1				
2270-2271 Compatible Test Set	8-15				
2270 Fixture Reports	8-25				
Modifying Test Set	8-9				
Nail Assignment Mode Options	8-16				
Nail Assignment Reports	8-20				
Preparing New Test Set	8-5				
Nail Types	8-3				

Item	TSPR	SOPR	SMGD	TLPR	TPRG
Networking					
Using GRnet		A-1			
Batch Operations		A-4			
Opens Testing					4-13
OVPs (Operational Verification Programs)	Refer to Maint Manual				
PIO (see Relay Driver Board)					
Power (Ac Line) Refer to Site Prep Guide					
Power Controller					
2271/2272		2-9			
2275		2-8			
Power Supplies (UUT)					
Configurations	4-12				4-51
Fixed Supplies	4-12				
Statements Inserted by ATG	7-28				
Test Language Statements	4-16				
HCS (High Current Supply)	4-15				
TPS (Triple Power Supply)	4-13				
UPS (Universal Power Supply)	4-15				
Wiring (Refer to Test Fixture Manual)					
Print Utility	3-20				
Printer (Strip)					
Controls (2271/2272)		2-7			
Controls (2275)		2-6			
Paper Loading (2271/2272)		3-10			
Paper Loading (2275)		3-7			
Printer (Combination)					
Controls (2275)		2-5			
Paper Loading (2275)		3-7			
Printer (Line)					
Controls (2271/2272)		2-12			
Controls (2275)		2-9			
Paper Loading (2271/2272)		3-11			
Paper Loading (2275)		3-8			
Ribbon Changing (2271/2272)		3-12			
Ribbon Changing (2275)		3-9			
Probe					
Description	4-23				
Connector Panel (2271/2272)		2-5			
Connector Panel (2275)		2-2			
SCRATCHPROBING		4-13			
Typical Connections (Refer to Test Fixture Manual)					
Purge Utility	3-21				

Item	TSPR	SOPR	SMGD	TLPR	TPRG
Receiver Connections (Refer to Test Fixture Manual)					
Relay Driver Board (PIO)					
Hardware Components	4-20				
Test Language Statements					4-55
Relay Driver Wiring	4-21				
Rename Utility	3-22				
Report Generator		5-15			
RSX-11/M					
MCR Commands			3-1		
System Software Structure	2-1				
Scanner					
Hardware					vi
Test Statements					6-79
SCRATCHPROBING					
2271/2272		4-13			
2275		4-11			
Setup Utility	4-29				
Shorts Testing					6-82
Software					
Overview (2271/2272)		1-10			
Overview (2275)		1-9			
Loading (2271/2272)		1-11			
Loading (2275)		1-9			
Software Structure	2-1				
System Software Cartridges		3-4			
SpeedPlus Option					
Difference Data	vii				
Hardware					4-35
Block Diagram					viii
Test Statements					4-37
Testing VLSI Devices	1-31			6-50	
Optional Features	4-28				
Fast Sections					3-14
Keywords					3-16
Test Devices					4-3
Timing Nails					6-102
BURST Parameters					6-105
SET CLOCK Statement					6-136
Split/Merge Utility	3-23				

Item	TSPR	SOPR	SMGD	TLPR	TPRG
Start-up/Shut-down					
System Start-up		3-5			
Aborting a Test (2271/2272)		4-15			
Aborting a Test (2275)		4-14			
System Shut-Down (2271/2272)		3-6			
System Shut-Down (2275)		3-5			
Emergency Shut-Down (2271/2272)		4-16			
Emergency Shut-Down (2275)		4-14			
STM (Self Test Module)					
Hardware					6-83
Test Statements					6-84
Strobe Pulses					
Description	4-23				
Connector Panel (2271/2272)		2-5			
Connector Panel (2275)		2-3			
System Manager's Data					
Overview			1-1		
Adding File Directories			4-8		
Backing-Up Source Files			2-4		
Changing Task Priorities			5-3		
Foreground/Background			6-3		
MCR Commands			3-1		
RSX-11 Utilities			3-11		
Login Accounts			4-1		
Indirect Command Files			3-14		
Partitions and Installed Tasks			5-1		
Checkpointing Strategies			5-2		
Dynamic Storage (Pool)			5-4		
Device Management			6-1		
Reloading Software (2271/2272)			2-1		
Reloading Software (2275)			2-4		
Loading 2294 DEVSYS Software			2-11		
Loading 2293 DEVSYS Software			2-14		
Setting Time and Date			6-5		
Software Updates			A-1		
Using ZAP			A-1		
Trouble Analysis			7-1		
Test Sub-Mode (Diagnose)					
2271/2272		4-20			
2275		4-18			
Testing					
Test Concepts	1-1	1-4			
Test Methods	1-3	1-4			
Testing Strategies	1-5				
In-Circuit Testing	1-4				
Functional Testing	1-7				
Connectivity Testing	1-18				
Analog Testing	1-19				

Item	TSPR	SOPR	SMGD	TLPR	TPRG
Testing (Cont'd)					
Digital Testing	1-25				
Breadboard Technique	4-6				
Testing Procedures (2271/2272)		4-1			
Testing Procedures (2275)		4-3			
SpeedPlus Testing	1-31				
Test Fixtures (Refer to Test Fixture Manual)					
2271/2272		1-10			
2275		1-11			
Test Language					
Format					2-1
Keywords					2-2
Statements (General					2-4
Statement Summary					B-1
Syntax					2-1
Test Program					
General	7-2				
Modification	7-35				
Preparation	4-2				
Analog Sample	7-22				
Digital Sample	7-32				
Test Set Preparation					
General	4-2				
Flow Chart	4-4				
Using a 229X System (Refer to System Manager's Guide)					
TEX Program	11-43				
TPS (Triple Power Supply)					
Hardware	4-13				
Wiring					4-51
Test Statements	4-18				6-88
Programming	4-20				
TRACS					
General		1-12			
Monitor Options		B-3			
Procedures		B-5			
Special Keys		B-9			
Error Messages		B-10			
Translate					
Running the Tranlator	9-1				
Translate Mode Options	9-2				
Error Messages	B-5				

Item	TSPR	SOPR	SMGD	TLPR	TPRG
Training Board (2275)					
Circuit	6-48				
Fixture (Refer to Training Board User's Guide)					
Difference Data	vii				
Tutorial Circuit	4-1				
Logic Diagram	6-47				
Fault Switches (Refer to Training Board User's Guide)					
Type Utility	3-25				
Unlock Utility	3-26				
Update Utility	9-24				
UPS (Universal Power Supply)					
Hardware	4-15				
Wiring					4-51
Test Statements	4-17				6-91
Programming	4-19				
VT10X Terminals					
2271/2272		2-13			
2275		2-8			
Keyboard Edit		11-2			
ZAP Utility			A-1		

GenRad

Production Test Division
300 Baker Avenue
Concord, Massachusetts 01742

USA
***Boston** • (Sales) Waltham, MA 02254
Tel: (617) 890-4900 • TWX: 710 324 0893
(Service) Concord, MA 01742
Tel: (617) 369-4400 • TWX: 710 347 1051
***Chicago** • 1156 Shure Drive, Arlington Heights, IL 60004
Tel: (312) 577-1881 • TWX: 910 291 1209
***Cincinnati** • Suite A, 2002 Ford Circle, Milford, OH 45150
Tel: (513) 831-9210 • TWX: 810 460 8323
***Colorado Springs** • 6180 Lehman Drive, Suite B207, Colorado Springs, CO 80907
Tel (303) 593-8238
***Dallas** • 1121 Rockingham Lane, Suite 100, Richardson, TX 75080
Tel: (214) 234-3357 • TWX: 910 867 4771
***Los Angeles** • P.O. Box 19500, 17361 Armstrong Avenue,
Irvine Industrial Complex, Irvine, CA 92714
Tel: (714) 540-9830 • TWX: 910 595 1762
***New York** • 22-08 Route 208, Fair Lawn, NJ 07410
Tel: (NJ) (201) 797-8001; (NY) (212) 964-2722 • TWX: 710 988 2205
***Orlando** • 3751 Maguire Blvd., Suite 170, Orlando, Florida 32803
Tel: (305) 894-4303 • TWX: 810 850 0270
***San Francisco** • 2855 Bowers Avenue, Santa Clara, CA 95051
Tel: (408) 727-4400 • TWX: 910 338 0291
***Washington, D.C.** • 1701 Research Boulevard, Rockville, MD 20850
Tel: (301) 424-6224 • TWX: 710 828 9783

CANADA
***Toronto** • 307 Evans Avenue, Toronto, Ontario M8Z 1K2
Tel: (416) 252-3395 • TLX: 06-967624
Montreal • Tel: (514) 747-1052 • TLX: 05-826652

EUROPE
***England** • Norreys Drive, Maidenhead, Berkshire SL6 4BP England
Tel: (0628) 39181 • TLX 851-848321
Best • Tel: (0) 49984240 • TLX 844-59535
***Milan** • Tel: (02) 8466541 • TLX: 843-320373
***Munich** • Tel: (089) 41690 • TLX: 841-529917
***Paris** • Tel: (01) 7970739 • TLX: 842-220991
Rennes • Tel: (099) 535154
***Rome** • Tel. (06) 455839
Wiesbaden • Tel (06121) 370057 • TLX 841-04186489
***Zurich** • Tel: (01) 552420 • TLX: 845-53638

ASIA
Japan • Tokyo Electron Limited, Shinjuku-Nomura Bldg.,
1-26-2 Nishi-Shinjuku, Shinjuku-Ku, Tokyo 160, Japan
Tel: 343-4411 • TLX: 232-2240 Labtel J
Other countries in Asia and Pacific • Tel: 344-9245 (Tokyo)

LATIN AMERICA
Contact WALTHAM, MA 02254 USA
Cable GENRAD Waltham (MASS.)

*Service Facilities Available

Printed in USA

See System Operation manual for Warnings and Cautions.

NOTE

The software package, described in this manual and supplied by GenRad, Inc. as a component of a system or as a separate item, is furnished for use on a single CPU. All proprietary rights are reserved by GenRad, Inc.

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in paragraph (b)(3)(B) of the Rights in Technical Data and Computer Software clause in DAR 7-104.9(a).

GenRad Inc., Concord, MA

The following are trademarks of GenRad, Inc.:

BUSBUST SCRATCHPROBING
GRnet SpeedPlus
NIM TRACS

NOTE

The material in this manual is for informational purposes only and is subject to change without notice. GenRad assumes no responsibility for any errors or for consequential damages that may result from the use or misinterpretation of any of the procedures in this publication.

Table of Contents

Section and Paragraph **Page**

FOREWORD

1 INTRODUCTION

1.1 General 1-1
1.2 Testing Concepts 1-2
1.3 In-Circuit Measurement and Isolation 1-18
1.4 Functional Testing 1-32

2 SYSTEM MONITOR

2.1 Introduction 2-1
2.2 Monitor Operation 2-6

3 FILE MANAGEMENT

3.1 Introduction 3-1
3.2 Specifying Devices 3-1
3.3 Specifying Files 3-7
3.4 Managing Files and Devices 3-9
3.5 Monitor Utility Descriptions 3-10

4 TEST SET ANALYSIS AND SPECIAL CONSIDERATIONS

4.1 General 4-1
4.2 Introduction of Tutorial Circuit 4-1
4.3 Outline of Steps in Test Set Preparation 4-2
4.4 The Test Plan 4-3
4.5 User-Prepared Input Files 4-8
4.6 Instrumentation 4-11
4.7 SpeedPlus Optional Features 4-26
4.8 Fixture Design 4-26
4.9 Setup Utility 4-29
4.10 Result of Test Set Analysis 4-32

5 AUTOMATIC TEST OPTIONS

5.1 Introduction 5-1
5.2 Writing Digital .ATO Commands 5-3
5.3 Writing Analog .ATO Commands 5-9
5.4 TRAINGBD.ATO 5-10

6 CIRCUIT DESCRIPTION PREPARATION

6.1 Introduction 6-1
6.2 Section Organization 6-1
6.3 CKTGEN Block Diagram Description 6-1
6.4 Preparing for CKTGEN 6-2
6.5 CKTGEN for the New User 6-2
6.6 CKTGEN Page Description 6-9
6.7 Detailed CKTGEN 6-27
6.8 CKTGEN Commands Summary 6-35
6.9 Command Mode 6-43
6.10 Circuit File Reader 6-44
6.11 Training Board Circuit Description 6-46
6.12 Circuit Description Language (CDL) 6-50

7 ATG PROCEDURES

7.1 Introduction 7-1
7.2 Run Circuit Description Compiler 7-4
7.3 Run Circuit Extractor 7-11
7.4 Generate Cross Reference Report (.AXR) 7-11
7.5 Run Analog Test Generation 7-16
7.6 Run Digital Test Generation 7-27
7.7 Merge Analog and Digital Program 7-35
7.8 User Modifications in ATG Programs 7-35
7.9 Automatic Test Generator (ATG) Mode 7-39

8 NAIL ASSIGNMENT

8.1 Introduction 8-1
8.2 Nail Types 8-3
8.3 Preparing a Test Set 8-5
8.4 Modifying a Test Set 8-9
8.5 Using 2270 or 2275 Test Set on Another System 8-13
8.6 Developing a Compatible 2270-2271 Test Set 8-15
8.7 Nail Assignment Monitor Page Options 8-16
8.8 Nail Assignment and Fixture Reports 8-20
8.9 2270 Fixture Reports 8-26

9 DEBUGGING THE TEST PROGRAM

9.1 General 9-1
9.2 Translating the Test Program 9-1
9.3 Debugging the Test Program 9-5
9.4 Diagnose Mode/Unlimited Access 9-12
9.5 Update Options 9-23

10 PRODUCTION TEST INFORMATION

10.1 General 10-1
10.2 Load Subroutine 10-1
10.3 Batch File Testing 10-1
10.4 Instruction Sheet 10-4

11 EDITORS

11.1 Introduction 11-1
11.2 Editor Utility Description 11-4
11.3 Directory of Editing Commands 11-6
11.4 Control Character and Escape Character Editing 11-10
11.5 GR Terminal Function Keys 11-15
11.6 VT10X Function Keypad Commands 11-19
11.7 DEC TECO Commands 11-25
11.8 ED2 Editor 11-31
11.9 GenRad TEX Program 11-43

APPENDIX A – DIGITAL LIBRARY DEVICES

Table A-1 A-1
Table A-2 A-4

APPENDIX B – SYSTEM ERROR MESSAGES

B.1 Introduction B-1
B.2 Circuit Description Compilation Errors B-1
B.3 Test Program Translator Errors B-5
B.4 Run-Time Diagnostic Major Errors B-9
B.5 RSX-11M System Error Codes B-12

INDEX

Foreword

PURPOSE OF MANUAL

This manual is directed primarily towards those manufacturing test personnel who are responsible for creating test software for testing pc boards on a 227X test system. It is organized by sections, to describe preparation activities, in the order that a test set preparer would normally need to address them while creating test software. These sections are:

Section 1 -- Introduction introduces the user to the concepts of circuit board testing. Discusses In-Circuit and functional testing and compares their interface and test result differences. Provides a complete description of In-circuit test techniques.

Section 2 -- System Monitor completely describes how to use the monitor. Shows how to make option selection on the menu-style page displays and how to initiate tasks.

Section 3 -- File Management defines the devices used to store files and defines the method of specifying files. The file structure of files used on the test system is listed. The utilities that are used to manipulate files are described.

Section 4 -- Test Set Analysis and Special Considerations discusses the various matters that need to be considered when starting to create a test set. Ways of handling unique or difficult circuit conditions are presented.

Section 5 -- Automatic Test Options Programming defines the programming requirements for user-desired entries in an ATG-created test program. Illustrates how to prepare the Automatic Test Options (.ATO) file.

Section 6 -- Circuit Description Preparation describes the menu-style displays used to create the Circuit Description (.CKT) file. Also describes the language used for the .CKT file and the method of manually creating it.

Section 7 -- ATG Procedures defines the processes that are part of the ATG process; namely, compiling the .CKT file, creating data base files, performing analog ATG, performing digital ATG, and merging the files. Illustrates how to select options for these processes.

Section 8 -- Nail Assignment describes the process of assigning test fixture nails to the test program and related files. Nail assignment options make it easy for the user to avoid conflicts on multiplexed systems.

Section 9 -- Debugging the Test Program describes the features provided for translating and debugging a test program and test fixture. Real-time debugging features and helpful display options are presented. Procedures for incorporating changes and updating all files to reflect changes are presented.

Section 10 —	Production Test Information indicates ways of communicating information about running a test between the test preparer and the test operator. Shows automatic methods of running tests using batch files.
Section 11 --	Editors. A reference of editing commands used by each of the editors available on the test system. Illustrates how to invoke an editor and how to use it. TECO and ED2 are the editors available.
Appendix A —	Digital Library Devices lists all digital devices that exist in the system library. The list is sorted by type designation and by function.
Appendix B —	System Error Messages. List of all error messages that the user is likely to see while compiling the .CKT, translating the .TPG, and using the editors.

RELATED DOCUMENTS

Other manuals that may be needed for information on creating and running test software are:

Test Programming manual
Test Library Programming manual
System Operation manual for the appropriate system model

Refer to the Document Directory for a complete list of manuals in the document set. The Document Directory is useful in finding information within the set of manuals.

DEFINITION OF TERMS

System	A term that refers to the workstation and related software and options.
Workstation	The term workstation may be used interchangeably with system in this manual.

DIFFERENCE DATA

Release 7.4 has several new features that improve the user interface to the system and that provide expanded testing capability.

CKTGEN

CKTGEN has a new library look-up ability, which means that digital device pin information is automatically inserted into the IC page. It also means that the user is told when no library routine is available for the device. The ability to define ICs not in a library has been improved with the addition of new IC pages.

Commands have been added to facilitate the entering of circuit data and to display circuit data already entered. Node connection data for any device or node can be requested at any time.

The ability to create a temporary .CIR file from the .CKT file makes it much easier to modify the .CKT file.

Refer to Section 6 for a complete description of CKTGEN.

SpeedPlus

GenRad 2271 and 2272 Workstations may be purchased or retrofitted with the SpeedPlus option. This option provides for a programmable clock signal of up to 20 MHz and driver/sensor test step rates of up to 5 MHz. Programmable clock synchronization is also provided for synchronizing to on-board clocks. Triggering is also provided for keying drive/sense timing to specific events. The clock and trigger statements are described in the 227X Test Programming manual. Synchronizing to on-board clocks is generally accomplished via commands in the .ATO file. Refer to Section 5 for these commands. Refer to the 227X Test Library Programming manual for instructions on writing SpeedPlus library routines. Refer to the 227X Test Programming manual for descriptionof SpeedPlus test statements. Digital library devices for which SpeedPlus routines have been added are indicated in Appendix A.

Special test hardware is required to provide clock and trigger signals and to provide high speed driver/sensor capability. This hardware must be wired to test points that are separate from normal test hardware. Therefore, nail assignment software is modified to accommodate these requirements. Refer to Section 8 for a description of nail assignment.

2275 Workstation

The addition of the model 2275 to the series of workstations requires a few new software descriptions. These include:

Disk device descriptions in Section 3
Disk utility pages in Section 3
Nail assignment report in Section 8
Emulation of 2275 test set to 2272 in Section 8

User Training Board

The 2275 is designed for operator self-training. To facilitate this training, a pc board and fixture are provided with each 2275, along with a self-training guide. This training board contains a variety of components which make it an excellent example circuit for use in this manual. Thus, the circuit is used extensively to illustrate test set preparation concepts.

Test Set Emulation

Nail assignment software has been upgraded to provide reports that reflect the receiver style of the 2275. It also provides for running the emulation mode to check for conflicts when 2275 test sets are used on a 2271 or 2272.

Introduction – Section 1

1.1 GENERAL

A 227X Board Test Workstation performs computer-controlled tests on circuit-board components. The circuit-board components are electrically accessed by a 'bed-of-nails' test fixture built specifically for each circuit board. Extensive programming tools supplied with the 227X can automatically generate test programs and instructions for wiring the test fixture, using circuit information entered by the user. The task that generates the test program is called the Automatic Test Generator (ATG). ATG uses a description of a circuit board, prepared by the user, and system test libraries to find out how to test each component and how to isolate it electrically from the circuit. It then writes a test program that switches appropriate measurement instruments to contacts in the test fixture and applies test patterns to each component on the board. When the test program runs, it returns information about each component that fails, identifying the component and the nature of the fault.

The 4 models of the test workstation are shown in Figures 1-1 and 1-2. The categories of testing implemented on and adaptable to a 227X Workstation are:

Circuit Board Continuity Tests
Analog In-Circuit Tests
Complex Analog Device Tests
Digital Component Tests
Digital Bus Tests
Functional Testing

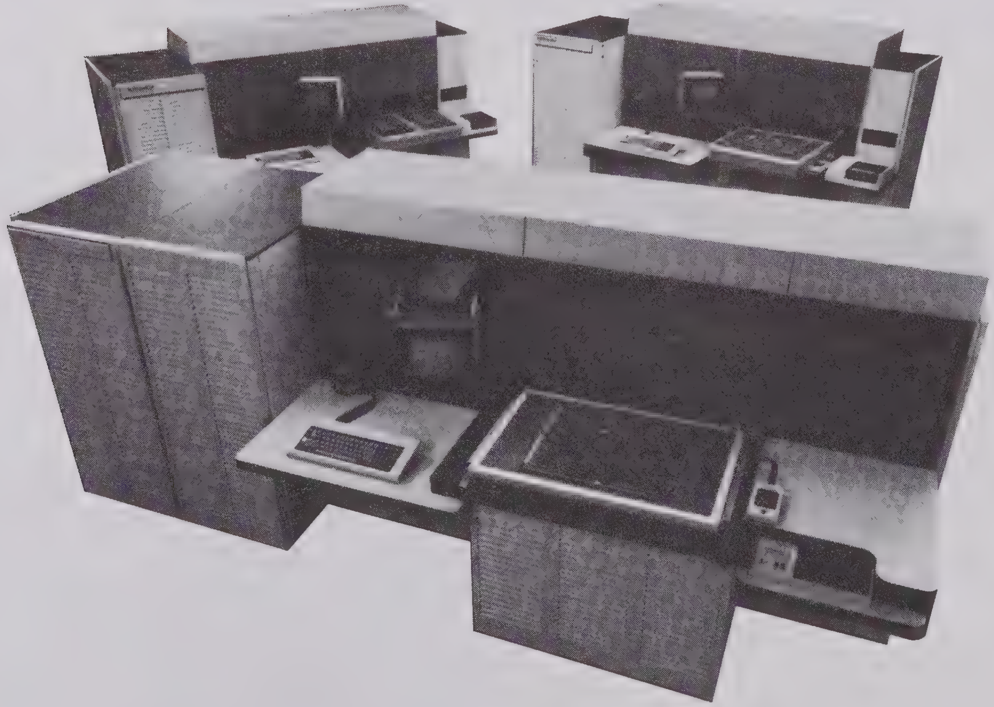


Figure 1-1. 2270, 2271 and 2272 Board Test Workstations.

The 2270 is best suited for testing low-pin-count boards with many analog components. It provides separate sets of digital, analog, and hybrid (digital and analog) test nails. The 2271 offers all the capabilities of the 2270 and can perform digital and analog tests of up to 960 nodes. All test nails have digital and analog capability. The 2272 is designed for testing large boards having between 900 and 3456 nodes. As with the 2271, both digital and analog tests can be performed on all nodes. The 2275 is a compact, low-cost system for testing smaller circuit boards requiring no more than 640 hybrid test nails.

On 2271 and 2272 workstations, optional test hardware is available for high-speed testing of complex digital devices. Called the SpeedPlus™ option, this enhancement provides up to 320 (2271) or 864 (2272) test nails that, in addition to standard digital and analog tests, can perform special digital tests at up to 5 million test steps per second. The SpeedPlus option also provides nails for separate clock signals and synchronization, a test triggering capability, and software for generating high-speed VLSI tests automatically.

The same test-development software is used on all 4 test systems, and programs developed on one system can usually be run on another if the required test hardware is available. Because the hardware differences have been made largely invisible to the user, this manual covers test development for all 4 systems.

1.2 TESTING CONCEPTS

1.2.1 General

The 227X Board Test Workstations are designed primarily for in-circuit component testing, although they can be used for functional testing as well. Both of these methods are commonly used in circuit board manufacturing, and it is important to have some understanding of each to gain the maximum utility from your test system.

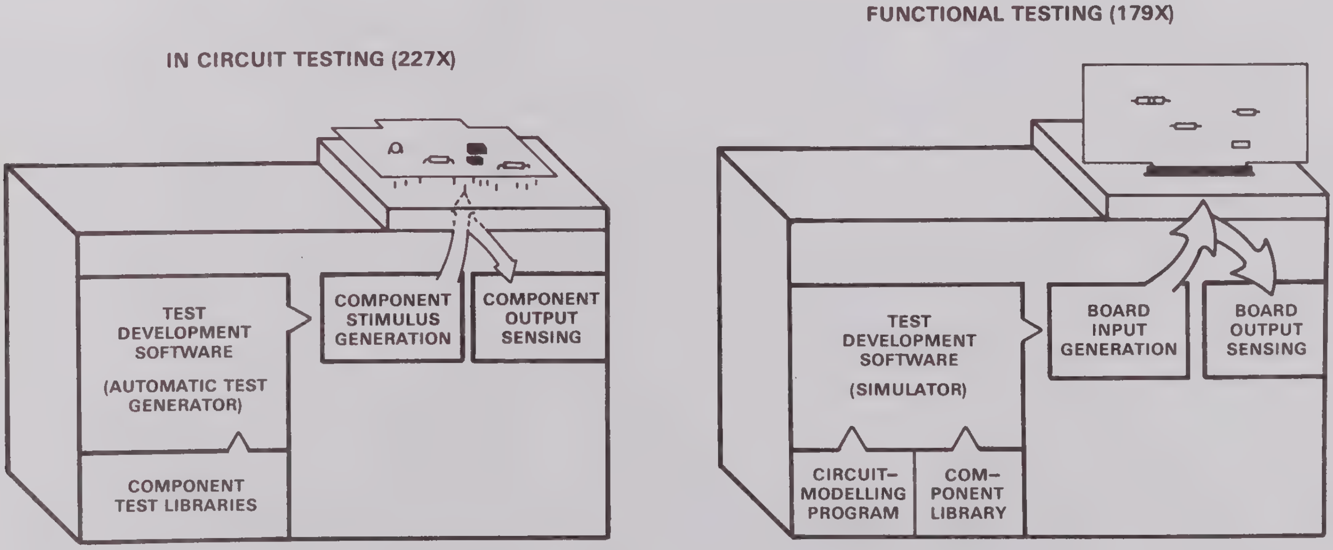
Figure 1-2.
2275 Board Test Workstation.

During manufacturing it is usually necessary to test for circuit board defects. The earlier a fault is discovered, the easier it is to repair it and return the board to further assembly. Most circuit board faults fall into 2 distinct categories that must be dealt with in separate ways. First, there are manufacturing faults, which may consist of isolated defects in components, in connections, or in the circuit board itself. A track on the board may be broken or it may be in electrical contact with another track because of a solder bridge or defective etch. Components may be missing; they may be inserted backwards, or their pins may be bent so that they cannot mate properly with a socket. The components themselves may be defective: resistors or capacitors may have the wrong values, inductors and transformer windings may be broken, transistors may have inadequate gain, digital ICs may fail to respond correctly to their inputs. Rigorously testing the parts before they are mounted on the board could reveal many component defects, but some faults only appear after the mechanical and thermal stresses of part insertion and soldering. Manufacturing faults will usually prevent a circuit board from functioning properly.

The second category of board defects includes those that result from the way the components function together in a circuit. If, for example, all the tolerance errors in the parts of an analog filter section accumulate in the same direction, the filter may exhibit performance faults, that is, it will not perform to its specifications, even though all the parts are within their specified tolerances. Likewise, a number of critical propagation delays in a digital circuit may add up to a delay that causes the circuit to malfunction, even though each of the components appears to function properly. To find these defects the board (or a functional section of a circuit board) must be tested as a whole by having the tester apply normal input signals and power to the board and check that the required outputs are produced.

1.2.2 Test Methods

These 2 distinct testing needs have given rise to 2 testing methods, shown in Figure 1-3, which can be used at various stages of circuit board assembly.


IN CIRCUIT TESTING (227X)

FUNCTIONAL TESTING (179X)

TEST DEVELOPMENT SOFTWARE
(AUTOMATIC TEST GENERATOR)
COMPONENT TEST LIBRARIES
COMPONENT STIMULUS GENERATION
COMPONENT OUTPUT SENSING
TEST DEVELOPMENT SOFTWARE
(SIMULATOR)
CIRCUIT-MODELLING PROGRAM
COM-PONENT LIBRARY
BOARD INPUT GENERATION
BOARD OUTPUT SENSING

Figure 1-3. In-Circuit and Functional Testing.

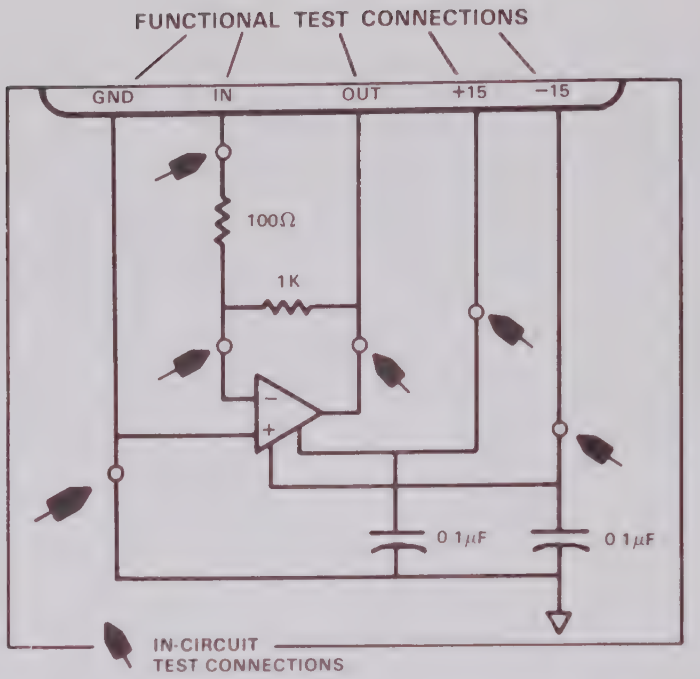


Figure 1-4. Simple Functional and In-Circuit Test Connections.

In-circuit testing methods are used to check for manufacturing faults, that is, for defects in individual components and interconnections on the board. To test components individually, connections must be provided from the tester to all functioning component pins. This requires a special test fixture that mates with each of the board's circuit nodes and test methods that can effectively isolate a single component from the parts surrounding it. An in-circuit component test is often performed as the first or second test after a board has been assembled and soldered. (It may be preceded by a separate test to check for opens and shorts in the conductive tracks before parts are inserted.)

The in-circuit test, however, cannot verify that the circuit board or its functional units perform correctly together. Hence, a functional circuit board test is often later used to find any faults not detected by the in-circuit test. With power applied to the board, this test checks that the board or chosen sections of the board produce the desired outputs when various input stimuli are applied, i.e., that the board or section performs as it should. It may detect faults that the in-circuit test could not identify, and it may also check that further stages of assembly have not damaged the board.

Figure 1-4 will help to clarify some of the basic differences between functional and in-circuit tests. This circuit board contains an operational amplifier in an inverting connection with a gain of 10. An input signal is fed from the edge connector pin marked IN, through the 100-ohm resistor to the amplifier's inverting input, and the output signal is supplied to the edge connector pin marked OUT. Power for the opamp is supplied to the edge connector at +15 and -15.

A straightforward method of testing the board is to wire a mating edge connector to ±15-V power supply voltages; a signal generator is then connected to IN and an ac voltmeter to OUT. After the board has been inserted into an edge connector, the supply voltages and a stimulus signal, such as a low-voltage sine wave, are applied. If the ac voltmeter registers 10 times the value of the input ac signal, we can assume that the board is functioning properly.

This is, in essence, the functional approach to board testing. Power is applied to the board, the whole circuit is tested, and the test instruments are attached only to the inputs and outputs from the board. The test does not concern itself with the internal operation of the circuit. In fact, the board shown in Figure 1-4 would probably still pass the functional test if, by mistake, 10-kilohm and 100-kilohm resistors had been inserted in place of 100-ohm and 1-kilohm resistors. (However, other, untested parameters, such as input impedance and noise, might not meet their specifications.)

The in-circuit approach, by contrast, attempts to check only individual components on the board. It therefore requires separate connections to each of the board's circuit nodes. These are usually made by spring-loaded, nail-like connectors in a special testing fixture which contact solder pads or tracks on the bottom of the board. Each of the circuit board's nodes is thus separately connected to the tester at the points shown by the nail symbols in Figure 1-4. An ohmmeter will be used to measure the resistance among all pairs of nails to ensure that there are no shorts on the board. The ohmmeter then measures the values of the 2 resistors through the nails that connect to them. A capacitance meter may check the values of the 2 capacitors. Finally, the opamp will be tested alone, in the open-loop configuration. Power will be applied across the opamp's power connections, a small positive voltage differential will be applied across its inputs, and the output will be checked to see that it has swung to within about a volt of the positive power rail. Applying a negative differential, the tester should then find the output near the negative rail. The test thus shows that the opamp IC has sufficient open loop gain to amplify the input voltage to a large output voltage and that it can drive the output pin in both directions.

While this test nowhere measures the closed loop gain of the amplifier circuit, we can be reasonably confident that if all the parts have correct values, the gain will have the correct value. Moreover, if any parts are of the wrong value, or are not properly connected, the in-circuit test can immediately identify which ones are faulty.

Both functional and in-circuit test methods are commonly used in circuit board manufacturing. In addition to these, it is useful to distinguish parametric test methods, which test for very specific characteristics of a component or a board. In the example above, for example, it might be important to know the gain-bandwidth product of the board, or the offset voltage of the opamp. Such measurements usually require special test preparation and may also require special instruments. To a limited extent, parametric measurements can be programmed by the user of the 227X; in general, however, the effects of surrounding components make it difficult to perform accurate parametric measurements on an in-circuit component. Critical component parameters are tested most efficiently by dedicated parametric testing equipment (such as the GenRad 1731 and 1732 IC test systems) at a stage prior to board assembly.

1.2.3 Overall Testing Strategies

With adequate hardware and supporting software, both in-circuit and functional test methods can produce thorough tests that find a high percentage of all possible defects, that is, tests with high fault coverage. Nevertheless, it will usually be found that one or the other method is easier to implement and provides more useful information for a particular stage of assembly. In-circuit tests are usually performed at earlier and intermediate stages of assembly, where it is most important to identify and correct component faults. Once it is established that all components are correctly inserted and are functional, a board-level functional test, or quality control test, can be useful for verifying a board's overall performance specifications.

While the in-circuit test provides no direct information about how well a board functions, experience shows that most boards that can pass an in-circuit test will pass a thorough functional test as well. Therefore, even though the in-circuit test does not normally determine whether the board as a whole works correctly, it can indicate whether a board should advance to further assembly or whether it has faults that need repair.

As soon as the in-circuit test finds a defect, the tester can produce a tag identifying the faulty component, so that the board can be returned for repair, usually without further testing. In addition, the tester assembles statistics about board defects, including such things as track shorts, open connections, and component failures. Periodically the tester can produce reports that list the causes of board failures, with their relative frequencies, so that design and manufacturing changes can be made quickly to improve the yield of usable boards. If additional networking facilities are provided, all board test information can be collected, processed, and compared at a central station to allow automatic, continuous monitoring of production and repair data (as in GenRad's TRACS system). After component defects have been found and repaired, interactive faults among the components (timing errors, crosstalk, amplifier offsets, combined errors of in-tolerance components) may still cause some boards to malfunction. Thus, a board-level (functional) test may be warranted at some manufacturing stage after the in-circuit test to verify the board's performance.

A dedicated functional test system (such as the GenRad 179X) is often used for this test. Alternatively, the user can expand the 227X into a test system with some functional test capabilities by adding optional power supplies and the IEEE-488 instrumentation bus for analog testing or by testing groups of ICs using the system's parallel driver/sensors. Functional testing on the 227X often allows the same test fixture to be re-used, and additional test statistics can be collected using the system's data logging feature. However, the Automatic Test Generation (ATG) programs cannot be used for generating a software model of the board. Instead, a test must be written in the tester's test language which applies an appropriate stimulus (a waveform or a bit pattern) at the board's inputs and measures the outputs to determine if the board or a group of components (such as a filter section) is functioning properly. If a sophisticated functional test is required, or if diagnostic information is to be gained from the functional test, some proficiency in manual programming in the test language may be required.

Because the ATG software and 227X hardware are optimized for in-circuit component testing, the test development procedures described in these documents apply primarily to in-circuit testing. Nevertheless, the most effective test strategies often combine some amount of functional with in-circuit testing, and it is therefore important that a 227X user have some understanding of functional methods as well. Both functional and in-circuit tests can be performed on the 227X tester, as well as many parametric tests.

The following paragraphs compare functional and in-circuit testing in greater detail, with particular attention to the test methods used by 227X Automatic Program Generation. The discussion will contrast a functional and an in-circuit test written for the circuit board shown in Figures 1-5 and 1-10.

1.2.4 Functional Methods

A functional test is, in a sense, most important to the final user of a circuit board because it certifies that the board meets its performance specifications, that is, that it performs the functions it was designed for. When a board fails, however, the functional test cannot always identify the cause of the fault immediately, since there can be any number of paths from the inputs to the outputs along which the fault might lie. As boards grow more complex, the possible signal paths, of course, grow much longer, and the simple information that a particular board output has failed grows less useful in identifying the cause of the failure. After a board fails a functional test, a technician will usually have to trace the fault from the output back to its origin, using a probe connected to an appropriate instrument, a signal tracer or logic analyzer, for example. If the tester itself is to aid in finding the fault, it must have a software model of the board's circuits available, so that it can compute the possible signal paths and guide the operator to probable sources of the error. (However, when a functional test is performed in conjunction with an in-circuit test, as described in para 1.4, the in-circuit test will often suffice to identify component faults that cause functional failures.)

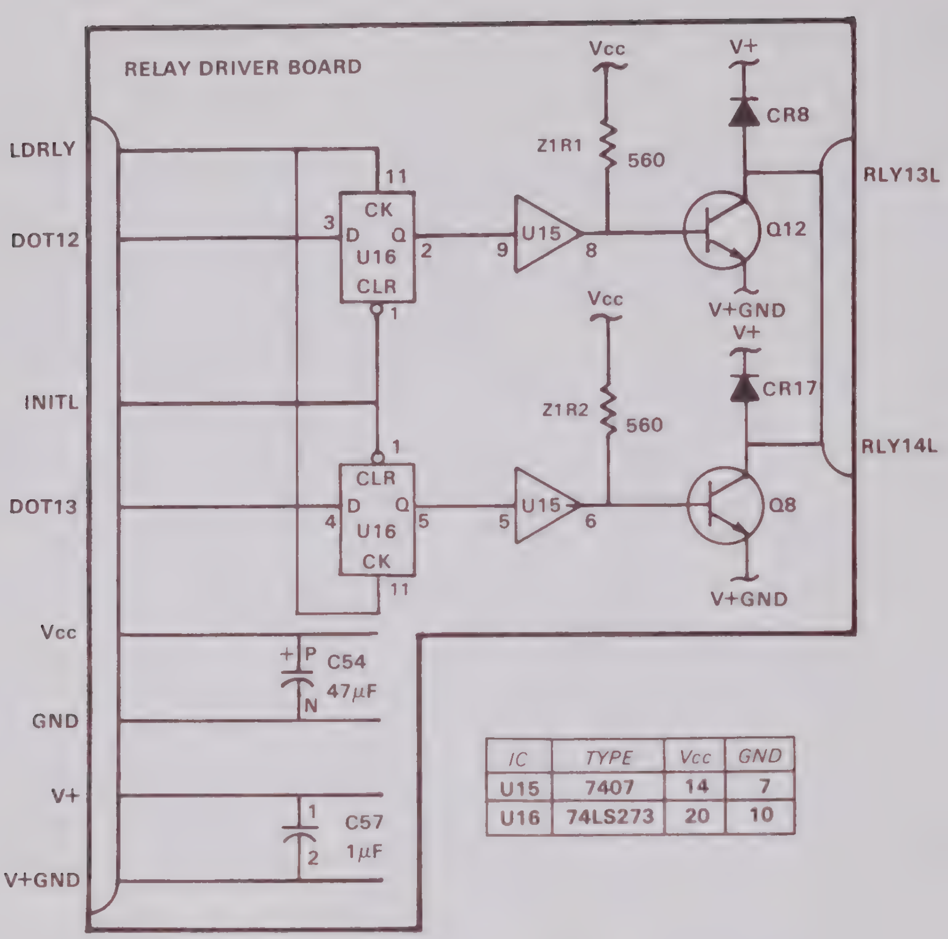
To connect the unit-under-test (UUT) to the test instruments, a fixture must be constructed that mates easily with the board and provides reliable electrical connections. Usually, only the board's inputs and outputs need to be tested, and the fixture needs to provide little more than the board's normal interface connectors. Therefore, the test fixture can be relatively simple and inexpensive to construct, which is especially attractive for low-volume testing. Using a functional tester, it is often possible to test a variety of bus-oriented boards from an edge connector in a single fixture.

Functional Test Example

For the sake of illustrating the kind of functional test that might be performed on the 227X, we shall describe a test that might be written for a simple relay-driver circuit board (shown in Figure 1-5). Bear in mind that all 227X test activities are directed by a computer that interprets commands written in the 227X test language, which is described in detail in the Test Programming manual. Actual implementation of a functional test on the 227X requires that the user write a program in the test language to control the tester's instruments and direct the accumulation of test results.

Consider briefly how the circuit board shown in Figure 1-5 functions and how its performance could be tested. The circuit is used as an interface between 4 relay control signals and the relay coils. When the signals DOT12 and DOT13 are applied, current must be drawn through the relay coils. The signals do not become effective, however, until they are clocked into the flip-flops of U16 by LDRLY. Each relay will be connected to a positive voltage, V+, and will close when Q8 or Q12 is turned on. Whenever the flip-flops are cleared by INITL, the relays will open.

It is evident that, to perform correctly, the board's outputs must be able to sink a specified current and that they must turn on and off correctly in response to the 4 inputs. A functional test, then, should verify a table of inputs and outputs, such as that shown in Figure 1-6, and it should check the on and off impedances at the outputs, to ensure that the proper currents will flow in the relay coils.


RELAY DRIVER BOARD
LDRLY
DOT12
INITL
DOT13
Vcc
GND
V+
V+GND
11
CK
3
D
Q
U16
2
CLR
1
9
U15
8
Vcc
Z1R1
560
V+
CR8
RLY13L
Q12
V+GND
Vcc
V+
CR17
Z1R2
560
RLY14L
1
CLR
D
Q
4
U16
5
CK
11
5
U15
6
Q8
V+GND
+
P
C54
47µF
N
1
C57
1µF
2

IC	TYPE	Vcc	GND
U15	7407	14	7
U16	74LS273	20	10

Figure 1-5. Sample Board for Functional Test.

Figure 1-6 shows a set of inputs and outputs, or test vectors, that can be used for the functional test. The first step of the test is to initialize the board by placing a low on the INITL line. Then, for INITL, DOT12, and DOT13, all the 3X3 possible input combinations are generated in Gray code, so that only one input pin changes state from one clock cycle to the next. Each new input pattern is clocked in by toggling LDRLY, and the expected outputs are tabulated for RLY13L and RLY14L.

The truth table of the board's inputs and outputs thus specifies a fairly complete functional test of the board, although not necessarily the best possible test. Careful examination of the circuit will reveal, for example, that the shaded test vectors are redundant; other possible improvements are beyond the scope of this presentation.

To perform the test, connections for inputs, Vcc, V+, and ground must be supplied at the left-hand edge connector. Connections for the outputs must be made at the right-hand connector. After applying power, the tester will apply the input pattern of highs and lows at TTL logic levels, as shown in the table. To test the outputs, the tester needs to check that the impedances -- not the voltage levels -- at RLY13L and RLY14L go from high to low, as required by the table. Most likely this measurement will be performed by connecting a pull-up resistor from each open-collector output to a voltage source; the outputs can then be checked by checking voltages rather than impedances.

Inputs				Outputs	
LDRLY	INITL	DOT13	DOT12	RLY13L	RLY14L
0	0	0	0	1	1
0	1	0	0	1	1
1	1	0	0	1	1
0	1	0	1	1	1
1	1	0	1	1	0
0	1	1	1	1	0
1	1	1	1	0	0
0	1	1	0	0	0
1	1	1	0	0	1
0	0	1	0	1	1
1	0	1	0	1	1
0	0	1	1	1	1
1	0	1	1	1	1
0	0	0	1	1	1
1	0	0	1	1	1
0	0	0	0	1	1
1	0	0	0	1	1

Figure 1-6. Functional Test Vectors for Relay-Driver Boards.

The value of the pull-up resistors must be carefully chosen. Notice that the 7407 is an open-collector buffer and that a high value of Z1R1 could supply too little base current to the driver transistor Q12, causing it to sink too little current from the relay. The same fault could arise if Q12's gain were too low. To detect this class of problem the external pull-up resistors and test thresholds must be carefully chosen to test the current sink capability of the transistors. The suppressor diodes at the outputs represent an especially difficult problem for a functional test. If CR8 or CR17 is missing, it is probable that an output transistor will be destroyed at turn-off by induced voltage from a relay coil. Hence, an additional measurement ought to be performed to check that the diodes conduct when the voltage at the outputs rises above V+. If the relays were mounted on the board, however, it might well be impossible to check for this kind of fault, since the tester would probably have no access to the protection diodes. There are other potential component problems, such as defective power supply capacitors, that the functional test will not detect.

Nevertheless, if the board passes the functional test, we can assume that it will work correctly when it is assembled into a system, even though the test gives us no certain information about individual components on the board.

If the board fails, however, this functional test can provide little information about the source of the fault. If, for example, the test finds that RLY13L never goes low (the transistor appears not to turn on), the cause could be a shorted diode CR8, a break in the connection between DOT12 and pin 3 of U16, or any of a number of other faults on the path between DOT12 and RLY13L.

The test therefore provides little information to identify the cause of the failure and to help repair the board. This results from regarding the circuit board as a 'black box,' with nothing but inputs and outputs. Of course, if an in-circuit test and test fixture are available (so that the tester can 'see' inside the board), they can be used to isolate a component fault after the board fails the functional test. Sophisticated functional testing, however, often takes a different approach. To produce a complete and efficient functional test, testers like the GenRad 179X test systems use a computer program called a simulator. The simulator draws on a library of circuit components to create a software model of how the board functions. The test programmer must write a series of input vectors for the test program, but the simulator predicts the expected outputs. It also predicts the effects of various kinds of faults and indicates how many of the possible faults the test can successfully identify. In addition, the tester can use the model to guide the operator to a suspected fault, since the model allows the tester to predict expected pin states at every point in the circuit. Starting at the output pin where the fault appears, the operator probes backwards through the circuit, node by node, until the failing node is passed. When it finds where the erroneous output signal originates, the tester can identify the source of the fault, and the board can be sent for repair.

The test vectors presented in Figure 1-6 for testing the relay-driver board were derived by listing all possible input combinations in Gray code and then examining the circuit to determine how the inputs should be used and what the outputs should be. With larger digital boards it often becomes impossible to write a thorough test in this way, since the potential circuit faults and the board outputs may be hard to predict. If a simulator is not available, an arbitrary set of inputs can be applied to a board that is known to be good, and the resulting output pattern can be recorded. This pattern, or signature, can then be matched against the outputs of boards being tested. But, lacking a model of the board, the tester cannot help the operator pinpoint the source of the fault.

To model a circuit, the simulator must be able to predict the state of each node as a signal advances from the inputs to the outputs. It must have available the complete truth table of each digital device on the board so that, for each input pattern that appears at an IC's inputs, it can find the resulting outputs and advance to the next devices. This has important consequences for functional testing of boards containing VLSI devices. For SSI and most MSI devices the truth tables can be recorded without difficulty, but for some VLSI devices, such as microprocessors, the truth tables comprise from perhaps 10,000 to many millions of test vectors. For these devices other, less thorough methods must be found for simulation. In general, adequate software modeling of boards containing VLSI devices is a formidable and expensive task.

We can summarize by listing the important characteristics of functional testing.

A functional board test may be defined as one that

Tests only inputs and outputs.

Makes connections through the board's own edge connectors.

Applies power to the board during the test.

Users of functional tests need to bear in mind that

- Significant manufacturing faults can escape detection.
- Board defects are usually found by operator probing.
- Test inputs must be prepared by hand.
- Thorough testing often requires a software simulator and component library.
- The library must model all devices as completely as possible.

1.2.5 In-Circuit Test Methods

Instead of examining a board's outputs with respect to a known set of inputs, an in-circuit test examines, individually, each component on the board. Thus, the in-circuit test provides no direct information about how well the board performs, but a well-designed in-circuit test can nevertheless identify most of the problems that will cause inadequate performance.

To access the components, the in-circuit tester requires a special kind of test fixture to provide electrical connections to all the component pins. This is usually a bed-of-nails fixture, specially prepared for the circuit board. It contains hundreds to thousands of spring-loaded contacts positioned to access each track (or node) on the board at least once. The test program is written for this unique test fixture, and the fixture must be designed to meet the needs of a particular testing strategy. Because the fixture and the test program are so closely related, the two together will be called a test set.

To avoid confusion, the word "nail" or "test nail" will be used only to refer to the spring-loaded connector in the test fixture that provides the tester's electrical connection to the unit-under-test (UUT). The word "pin" is reserved for component leads or interconnections (thus, an IC may have 16 pins, and a capacitor is said to have 2 pins).

In-circuit tests generally fall into two categories: analog tests to measure component values, and digital tests to check the operation of digital integrated circuits. Test hardware appropriate to each kind of test is provided in the system, but it must be connected to each part on the board individually before a digital or analog test can be performed. In addition, an in-circuit test usually begins with a set of analog measurements called a connectivity test. This test checks for shorts and open connections on the board.

As shown in Figure 1-7, a circuit board is tested by pressing it against spring-loaded test nails in the test fixture (usually by means of a vacuum applied on the contact side of the board). Since there may be thousands of test nails making contact to the circuit board under test, it would be impractical to provide measurement instruments for each test nail. Instead, each of these nails is wired to a number of fast, low-resistance relays in a multiplexer called the scanner. For analog testing, each nail can be connected to any of four lines (or 'channels') in the analog test bus (horizontal connections in Figure 1-7) by selectively closing and opening the relays. These lines, in turn, can be connected to measurement instruments via a second relay matrix, called the instrument multiplexer. Thus a single measurement instrument may be connected through as many as 4 lines to any set of nodes on the UUT.

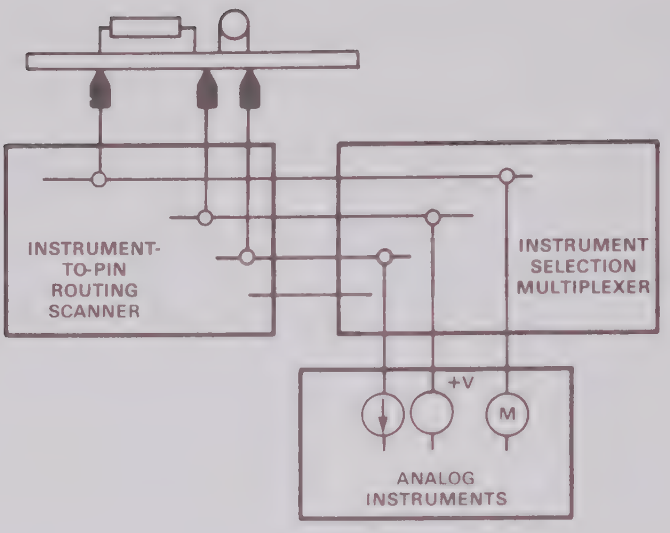

INSTRUMENT-TO-PIN ROUTING SCANNER
INSTRUMENT SELECTION MULTIPLEXER
+V
M
ANALOG INSTRUMENTS

Figure 1-7. Analog Test Hardware Connections.

The test instruments used in digital testing are called driver/sensors. They can place logic levels on the test nails or sense logic levels, according to voltage, timing, and other parameters defined in a test program. Digital inputs are usually connected to other digital outputs, as shown in Figure 1-8, which means that the tester may need to use the same node to drive the inputs to one IC and to sense the outputs of another. The driver/sensors are thus equipped to operate either as digital pulse generators or as logic sensors.

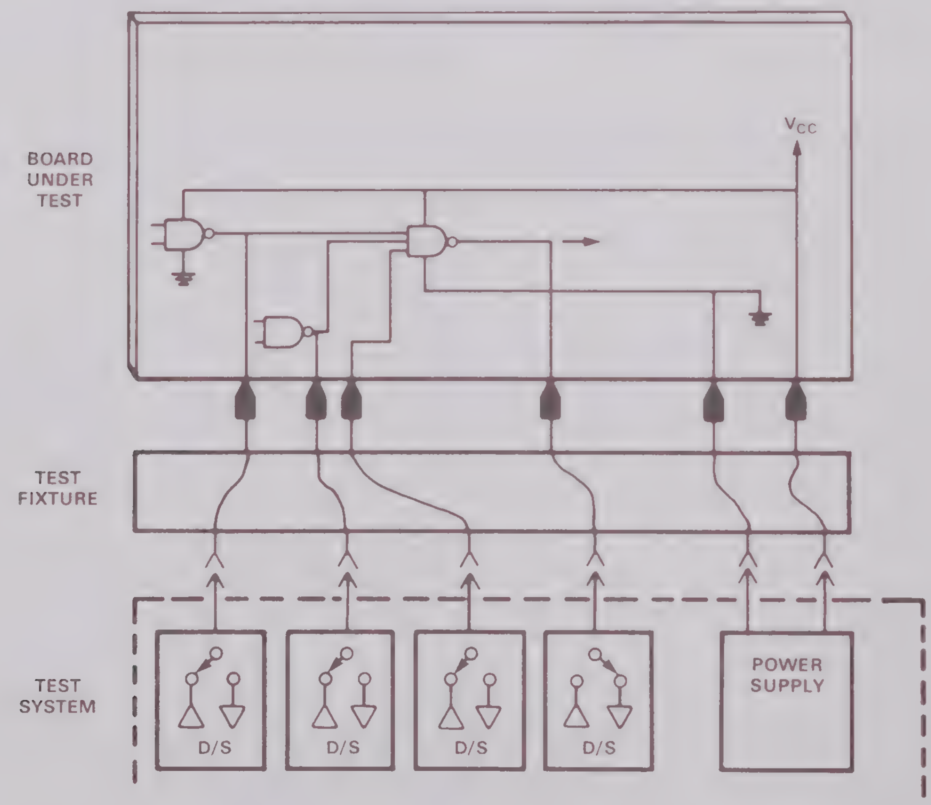

BOARD UNDER TEST
V_{CC}
TEST FIXTURE
TEST SYSTEM
D/S
D/S
D/S
D/S
POWER SUPPLY

Figure 1-8. Digital IC Test Hardware Connections.

A digital IC is tested by applying power to the board and input bit patterns to the IC; the outputs are checked for correct logic levels. The test must often also control adjacent ICs to prevent them from interfering with the test. Because a digital test must often control a number of ICs at once, digital testing usually requires a large number of driver/sensor channels. Tests of bused devices, such as microprocessors, may require many more driver/sensor channels than the IC alone would require.

In addition to special fixturing, in-circuit testing also requires special software and test techniques. Components on a circuit board are rarely electrically isolated; instead, they are usually interconnected. Each test for a component must make an accurate measurement, in spite of the effects of the parallel connections of other components to the same circuit nodes. The test development software must analyze the circuit environment of each component and then design a test that will be able to ignore these effects. In analog testing the software must isolate, or 'guard,' the component under test and must also account for the resistances of the test nails, multiplexing relays, and wires that come between the device-under-test and the measurement instruments. To test digital ICs, methods must also be found to make the ICs work as if they were isolated components, without connections to other ICs or buses.

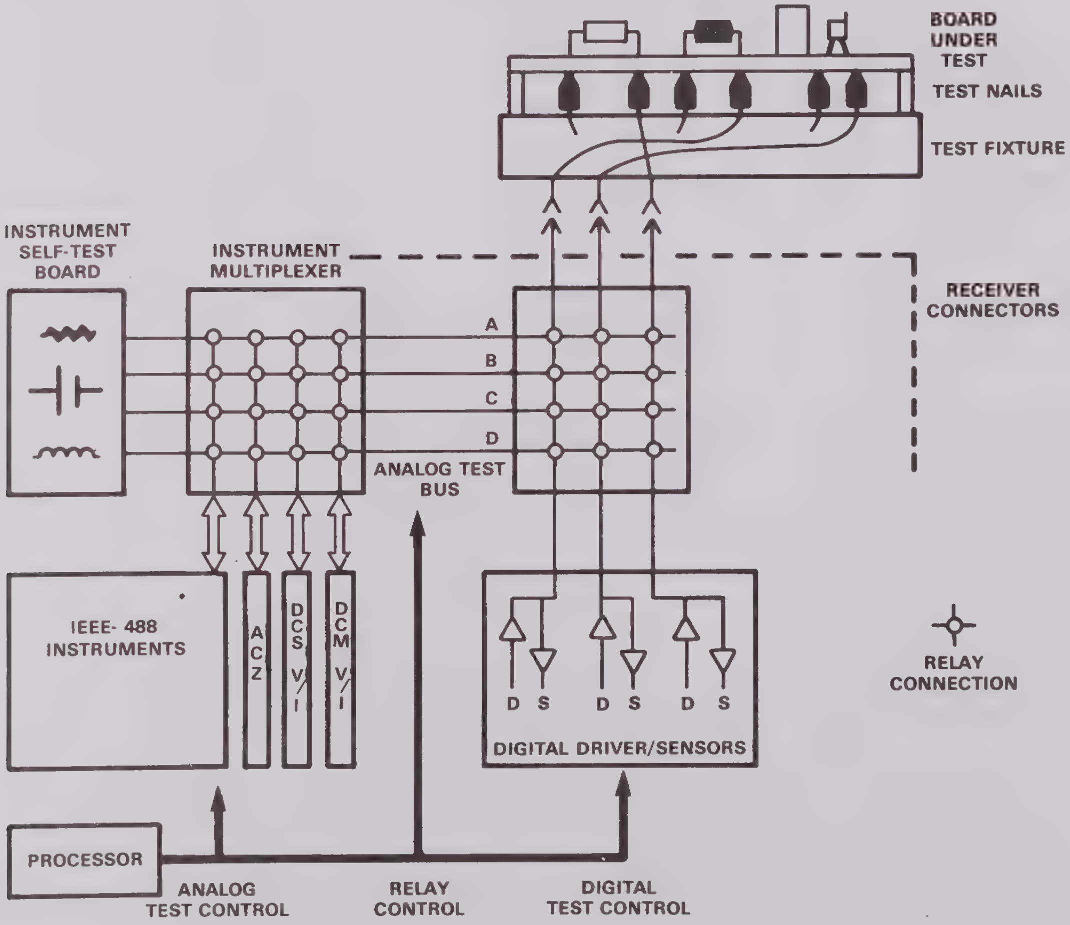

BOARD UNDER TEST
TEST NAILS
TEST FIXTURE
INSTRUMENT SELF-TEST BOARD
INSTRUMENT MULTIPLEXER
RECEIVER CONNECTORS
A
B
C
D
ANALOG TEST BUS
IEEE- 488 INSTRUMENTS
ACZ
DCS V/I
DCM V/I
D S
D S
D S
DIGITAL DRIVER/SENSORS
RELAY CONNECTION
PROCESSOR
ANALOG TEST CONTROL
RELAY CONTROL
DIGITAL TEST CONTROL

Figure 1-9. Block Diagram of System for Analog and Digital In-Circuit Testing.

Thus, an important component of both analog and digital in-circuit testing is a computer that can analyze a circuit and develop appropriate tests. Then the computer must set up each digital or analog test by closing relays that connect the required instruments to test nails in the test fixture. It also sets up the test instruments and processes the measurements that they return. As shown in Figure 1-9, the test nails can be connected to both digital and to analog instruments, so that processor can, if necessary, carry out both an analog test (such as an impedance measurement) and a digital test (for correct digital functioning) on each component in the circuit. Because the test nails can be used for both kinds of testing, they are called hybrid nails. All test nails in the 2271, 2272, and 2275 systems are hybrid nails. The 2270 system, however, can also be provided with digital-only and analog-only test nails. The digital nails are connected only to digital driver/sensors, and the analog nails are connected only to the scanner. The SpeedPlus option, available on 2271 and 2272 systems, provides some additional nail types. The SpeedPlus nails include hybrid driver/sensors that can operate as standard hybrid nails or as special, high-speed digital nails. There are also special clock and synchronization nails that facilitate testing complex VLSI devices.

The overall accuracy and success of in-circuit analog and digital testing depend on a number of carefully coordinated strategies. These include proper test sequencing, as well as techniques for isolating analog components, disabling buses, disabling or inhibiting of devices surrounding a digital IC to ensure its isolation, identifying/breaking feedback loops, parallel testing of groups of devices, and simplified probing to verify fixture-to-device connections.

An in-circuit testing sequence typically begins by checking all circuit paths for unwanted opens and shorts. The test system reports these as board defects; in addition, it uses the information later in the test sequence to skip tests of nodes and components that could be affected by them. This prevents harm to components when power is applied and reduces the chance that these circuit board faults will be diagnosed as faulty components later on.

Next, discrete analog components such as resistors, capacitors, inductors, diodes, and transistors are tested by measuring characteristics like resistance, capacitance, or forward voltage drop, with no power applied to the board. Each device is checked for correct insertion and orientation, proper value, and connection to the circuit nodes. If active components, such as operational amplifiers, are present, these are tested after the passive components, with power applied.

Once analog testing is complete, power and ground are applied to the board and each digital IC is isolated and tested individually for proper functional operation. The sequence of digital testing is organized so that non-bused ICs are tested first, using functional test patterns drawn from a library of IC tests. Techniques called backdriving, disabling, and inhibiting are used to make the IC appear to the tester as if it were functioning in isolation, without connections to other components on the board. Thus it is only necessary to apply an appropriate set of input bit patterns to the IC and to check that it produces the proper output patterns.

Bused ICs present a special class of problems. Before full tests are performed on bused devices, all the devices on the bus are tri-stated, i.e., put in a state where they are not driving the bus, to verify that the bus is free.

If the bus test fails, GenRad's BUSBUST diagnostic technique can identify automatically the IC(s) causing the bus failure. Once the bus test passes, the performance of each IC on the bus can be individually tested. All other devices on the bus are disabled, that is, disconnected from the bus, and the IC is tested as if it were the only device on the board.

Thus, in contrast to the functional test,

Components are tested rather than entire board.

Test connections are made through a special fixture.

Power is not applied, except for active components.

Because the tester examines each component individually, it can usually localize a fault immediately and issue a report telling the operator or technician which part needs repair. Interactive faults that can cause the board to malfunction will probably not be noticed, but such faults will be rare on a well-designed board. The most troublesome aspect of the test is the assumption that each component can be tested as if it were the only part on the board. Fairly complex test strategies must be devised to analyze the component's circuit environment and to isolate the component from the surrounding circuit. However, since these test strategies involve only individual components and their immediate circuit connections, libraries of standard test procedures can be assembled and adapted to the limited range of environments in which the components are found. Given an appropriately coded description of the circuit, system software can write the entire board test by drawing tests for individual components from a test library, and VLSI devices are tested in a way that is, in principle, no more complex than the method used for simple components.

The system has no need to know the entire truth table or transfer function of a digital device, since it does not analyze how signals move through the circuit. Instead, it only requires that the library contain for each device a set of typical input and output patterns that it can use to test the device. For simpler digital ICs, these patterns will probably be based on the devices' truth tables, but for larger devices, they may consist of no more than a carefully chosen sample of the possible inputs and outputs.

Thus, while the functional tester's simulator requires a complete model of a digital device, the in-circuit tester requires only enough information to write a reasonably effective test for the device. In addition, it usually needs a set of device inputs that can be used to disable the IC while other neighboring ICs are being tested.

We can summarize by listing the important characteristics of in-circuit testing. A full comparison with functional test methods appears in Table 1-1.

Identifies board defects immediately.

Detects most manufacturing faults.

Requires circuit analyzer, test generator, and library.

Provides system software to write and evaluate the test.

The library contains model tests rather than complete transfer functions.

Because each component on the board can be tested individually, the in-circuit tester can usually identify a component fault at once, with little or no operator probing. For those cases where a fault such as an open connection needs to be localized further, GenRad's Scratchprobing technique is available for optional use. It allows the tester to discriminate between a poor test nail contact, a bent IC pin that was not inserted correctly, or a broken track.

Table 1-1
FUNCTIONAL AND IN-CIRCUIT TEST CHARACTERISTICS

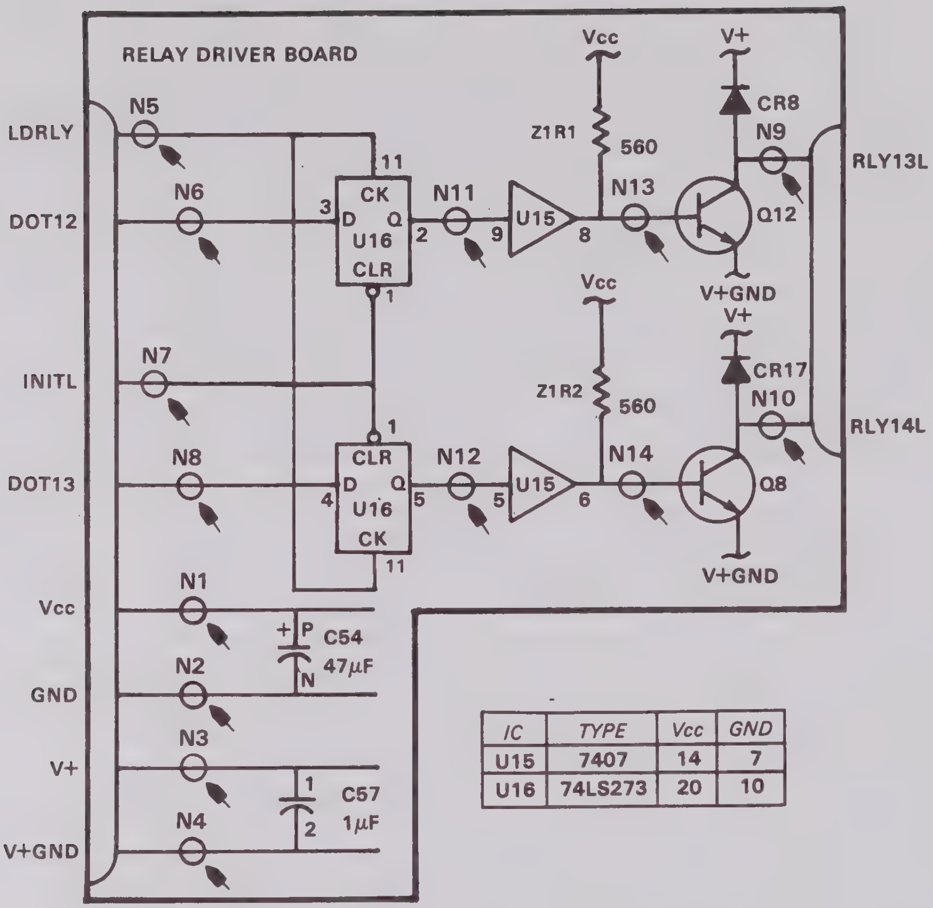
	Functional (Board-Edge) Test	In-Circuit Test
What Tested	Only inputs and outputs	Only component connections, values, functions
How Tested	With power applied to board	Component-by-component; power applied only when testing active components
Fixture Requirements	Board edge connector, may accept many boards	Expensive bed-of-nails for each board design
Software Requirements	Circuit simulator to predict outputs, fault coverage, aid diagnosis; simulator library; user writes at least part of test	Automatic Test Generator writes test, using library of test procedures; less programmer expertise required
Component Library	Must contain complete transfer function or truth table for each digital device	Contains model tests, written for variety of circuit environments; contains analog and digital tests
Fault Diagnosis	By guided operator probing; fault identification depends on adequate simulation	Usually requires no probing; Good fault identification
Fault Coverage	Up to 90 % of board faults found (depends on board design)	Around 90 % of board faults found (depends on board design)

In-Circuit Test Example

More specific descriptions of in-circuit isolation and measurement techniques follow in para 1.3, and the Automatic Test Generation software, which is normally used for writing in-circuit tests, is introduced in Section 7.

For the sake of comparison with the functional test of the relay-driver board of Figure 1-5, we shall describe briefly how an in-circuit test would be prepared for use with the 227X test system. Again, the actual implementation of the test would be a program written in the test language. In most cases the user will not need to write the test but will, rather, let the system software prepare it from a description of the circuit.

The relay-driver board was tested functionally by applying power and then checking the outputs against a table of outputs and inputs. For an in-circuit test of the board's components, connections must be provided to the 14 circuit nodes labeled N1 to N14, shown in Figure 1-10. The system then tests


RELAY DRIVER BOARD
LDRLY
DOT12
INITL
DOT13
Vcc
GND
V+
V+GND
N5
N6
N7
N8
N1
N2
N3
N4
CK
D
Q
U16
CLR
11
3
2
1
4
5
N11
N12
9
U15
8
6
Z1R1
560
Z1R2
560
Vcc
V+
CR8
N9
N13
Q12
V+GND
CR17
N10
N14
Q8
RLY13L
RLY14L
+
P
C54
47µF
N
C57
1µF

IC	TYPE	Vcc	GND
U15	7407	14	7
U16	74LS273	20	10

Figure 1-10. Sample Board for In-Circuit Test.

individual components by connecting appropriate measurement instruments to each component in turn. For the analog portion of the test, the test program will specify which relays must be closed to attach test instruments to each component before it is tested. (Where relays are used in digital testing, the system closes the necessary relays automatically.)

As mentioned above, an optimal test is created by performing the component tests in the correct sequence. The first test will therefore be for shorts and opens; it will check all possible pairs of nodes for low or high resistance, using a test voltage below the lowest diode threshold (so that all semiconductor junctions will appear to the tester as open connections). For example, the tester will measure the resistance from N5 to N6; if it finds a very low resistance, it will report a short from N5 to N6.

Next, passive analog components such as C54 and CR8 will be checked. For C54, a capacitance meter will be connected to N1 and N2 by closing appropriate relays, and, if the measured value falls within a specified tolerance range, C54 will pass. To test CR8, the tester connects a programmable current source and a dc voltmeter from N9 to N3 (V+). While directing a small current from N9 to N3, it checks that an acceptable forward voltage drop appears across the diode. In addition, the test may check the diode for reverse-bias current, and it can check the capacitors for proper values (at least as far as the components connected in parallel with them do not interfere). The in-circuit test can thus identify a missing or faulty protection diode in the signal path, as well other individual component faults which the functional test could not identify.

To test the digital ICs, the tester powers up the board by connecting a power supply across N1 and N2. It performs functional tests of the 2 flip-flops, driving all input pins high and low and checking that the output lines (N11 and N12) respond correctly. While U16 can be tested as if it were the only device on the board, U15 presents potential testing problems because of its connections to U16. The flip-flops' outputs cannot be disconnected from U15; they will continue to drive the inputs in parallel with the tester during the test. Thus, if one of these outputs happens to be high when the tester needs to supply a low to an input of U15, it must override the flip-flop by supplying enough current to pull it briefly to the desired state. This isolation technique is called backdriving. The flip-flops' clock input presents an additional problem. Digital noise from the test of U15 could conceivably cause the flip-flop to change state during the test, briefly giving a false output from U15 despite the tester's attempt to backdrive the flip-flop's output. Therefore the test will invoke another isolation technique, called inhibit, to hold the flip-flop in a known state. From the system IC library the tester will find that it can hold the flip-flops' output stable by pulling the CLR line low. Since CLR overrides all other inputs, the outputs cannot change state and cannot disturb the testing of U16.

At this point the performance of every component on the board will have been verified, as well as the continuity of all connections between components. If faulty components or connections were found, the tester issued a report naming the faulty component, and it added information about the failure to its file of statistics for the board type. These statistics are as important a part of the test as the individual board results, as they provide information that can be used to lower the rate of board failure -- in essence, to repair the board before it is tested.

While the component tests performed on the relay-driver board are representative of the kinds of in-circuit tests commonly performed, they were chosen for their relative simplicity. Typical tests require considerably more analysis of each component's circuit environment and evaluation of possible test methods.

1.3 IN-CIRCUIT MEASUREMENT AND ISOLATION

The in-circuit tests generated by the 227X Automatic Test Generation (ATG) software generally consist of three distinct sections: connectivity tests, analog component tests, and digital tests. Simple examples of these tests were provided in para 1.2.5. The following paragraphs provide more detailed descriptions of the measurement techniques used in each of these sections. Users who need to modify tests created by ATG or who need to write tests of their own should thoroughly understand these techniques and the criteria ATG uses in selecting various test methods.

1.3.1 Connectivity Testing

When ATG creates an in-circuit test for a circuit board, it first writes a set of tests for shorts and opens in the circuit board connections. These general connectivity tests are carried out by measuring resistances between all possible pairs of nodes. Unconnected nodes are tested for very high resistance; connected nodes are checked for very low resistance.

Another, special form of connectivity testing, the Scratchprobing diagnostic technique, may be used later in a test to determine if a component failed because of a bad connection on the pc board. A probe is used for this test. When a component fails, the operator can be instructed to drag the probe across the pins of the failed component. This tests the connection from each IC pin to the test nail to which it is connected. It allows the tester to find out whether or not the connection from the test nail to the IC pin is intact and, thus, whether the IC or the board connection needs repair.

1.3.2 Analog In-Circuit Testing

Two essential problems must be solved in each in-circuit analog test. First, a relevant measurement parameter, such as resistance, capacitance, or gain, must be defined and a stimulus current or voltage must be selected for measuring it. Second, means must be found to isolate the device from its circuit environment, so that interactions from other components will not affect the measurement. In addition, the test must issue relay commands to connect the appropriate circuit nodes to the tester's measurement instruments. In difficult cases, both analog and digital test strategies to accomplish these ends can become quite complex. Fortunately, most testing problems can be solved by the 227X's Automatic Test Generation (ATG) software, using programmed circuit analysis and libraries of test procedures for circuit components to develop tests for a board that has been described to it in a specially coded circuit description file.

When ATG is used, the system software automatically analyzes the circuitry surrounding each analog component and then searches the Analog Test Library (ATL) for an appropriate measurement configuration. Using the library procedure as a template for the test, it chooses source, measure and guard terminals, calculates the unknown component value, and checks for error conditions. Tests for most basic analog devices are contained in the Analog Library and are automatically selected by the ATG task. Descriptions of these tests are provided in the 227X Test Library Programming manual. Information on how to write tests not generated by ATG appears in the Test Programming manual.

Analog testing is usually performed with no power (or ground) applied to the UUT. Passive analog components are tested by applying an external stimulus (either voltage or current) to the component and measuring the results. Tests of more complex devices, such as active filters, may require that power be applied. In either case appropriate source and measure instruments are connected to component pins on the board, forcing a voltage or current input, and a resulting current or voltage is measured.

The test selection process can generate 2 types of basic impedance test configurations. These are classified as follows:

2-Terminal Unguarded Measurement
4-Terminal Guarded Measurement

Other measurement configurations, including 4-terminal Kelvin measurements, can be programmed by the user.

2-Terminal Measurements

When circuitry surrounding a part allows, ATG will choose a 2-terminal test. As shown in Figure 1-11, a 2-terminal measurement is performed by applying a known voltage to the component from a programmable voltage source and measuring the resulting current to a 'virtual ground' formed by the operational amplifier. This measurement scheme is basic to most of the 227X's analog measurement techniques.

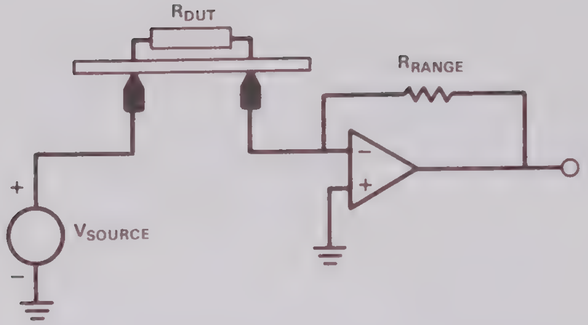


Figure 1-11. 2-Terminal Resistance Measurement.

The virtual ground is provided by an operational amplifier that serves as a current meter. The opamp always sources or sinks enough current to Rrange to hold the inverting input within much less than a millivolt of the grounded, non-inverting input. Applying a voltage to the device under test (DUT) causes a current to flow through the DUT to this virtual ground, and the size of the current can be calculated from Vmeas, the voltage the opamp must produce to hold its inverting input at ground.

While this configuration has the advantage of requiring only 2 test nails, it cannot isolate the DUT from the effects of other components that may be connected in parallel with it. The voltage applied to the DUT may find shunt paths to virtual ground (that is, the measure node) through these adjacent elements. If these currents are not taken into account, the measured value of the component will be too low.

In addition, as Figure 1-12 shows, a more exact view of the measurement set-up must include the resistances of the wires and relays that come between the test nodes and the instruments. These typically contribute 1 to 2 ohms of total residual resistance between the source instrument and the DUT, 500 to 1000 pF of capacitance to system ground, and around 20 uH of series inductance during analog measurements. (These residual values apply only during analog testing; for digital testing the scanner lines are usually disconnected. Refer to system specifications for residual values applicable to your system.)

If ATG's circuit analysis shows that the required measurement accuracy cannot be obtained with a 2-terminal test, it will try to write a test using the guarding technique described in the next paragraphs.

4-Terminal Guarded Measurements

Components are typically surrounded by other components that provide shunt paths, as shown in Figure 1-13. The effect of these can be temporarily eliminated by using an electrical isolation technique called guarding. A

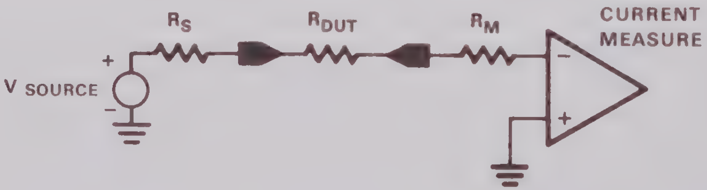


Figure 1-12. Series Residual Impedance Error.

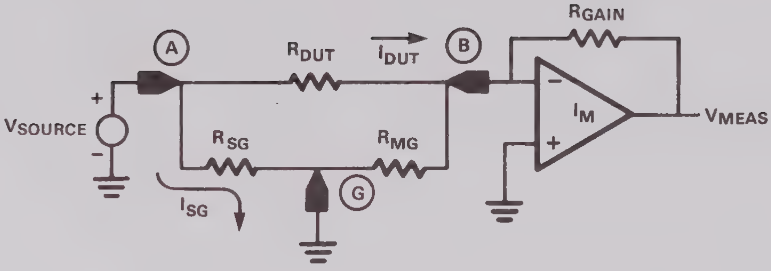


Figure 1-13. 3-Terminal Guarded Measurement.

component under test is guarded by applying a voltage equal to the voltage at the measure node (B) to all nodes adjacent to the component pins to prevent current flow in the shunt paths from reaching the measure node.

To guard the resistor Rdut, all nodes electrically adjacent to it (excluding the component nodes) must be connected together and tied to ground (0 V), which is the system guard voltage. Next, a voltage is applied to node A and the resulting current measured at node B through the instrument arrangement shown in Figure 1-13.

Rsg and Rmg represent the equivalent impedances from nail A to guard and nail B to guard, respectively. All the current through Rsg returns to ground via the guard nail G. Because the current meter input is held at ground potential, both ends of Rmg are effectively grounded, and no current can flow through it. Therefore all the current that flows into the current meter flows through Rdut, and the current flowing in the shunt resistances has been effectively screened from the measurement circuit.

Dividing the voltage set by Vs by the current measured by the ammeter, the 227X software obtains a measured value of Rdut and compares it against the tolerance limits for the resistor. If it is within the specified tolerance limits, the device passes; otherwise, it fails.

While usually more accurate than ATG's 2-terminal measurement, this guarded measurement is also susceptible to errors. We have assumed that nails B and G are at the same potential. However, the ground connection at G actually has a small amount of resistance, typically the 1- to 2-ohm residual resistance represented by Rg in Figure 1-14. If a significant current flows into nail G, the node that it contacts will actually rise above ground potential. If, in addition, Rmg is small relative to Rdut, a significant error current, Ie, can flow from it into the current meter. Hence, the ratio Rdut/Rmg, called the guard ratio, is important for the accuracy of a guarded measurement. If it is

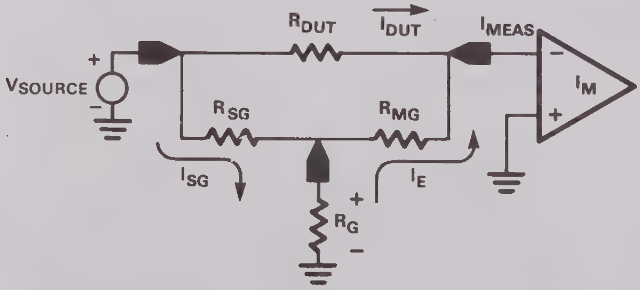


Figure 1-14. Guard Path Impedance Error.

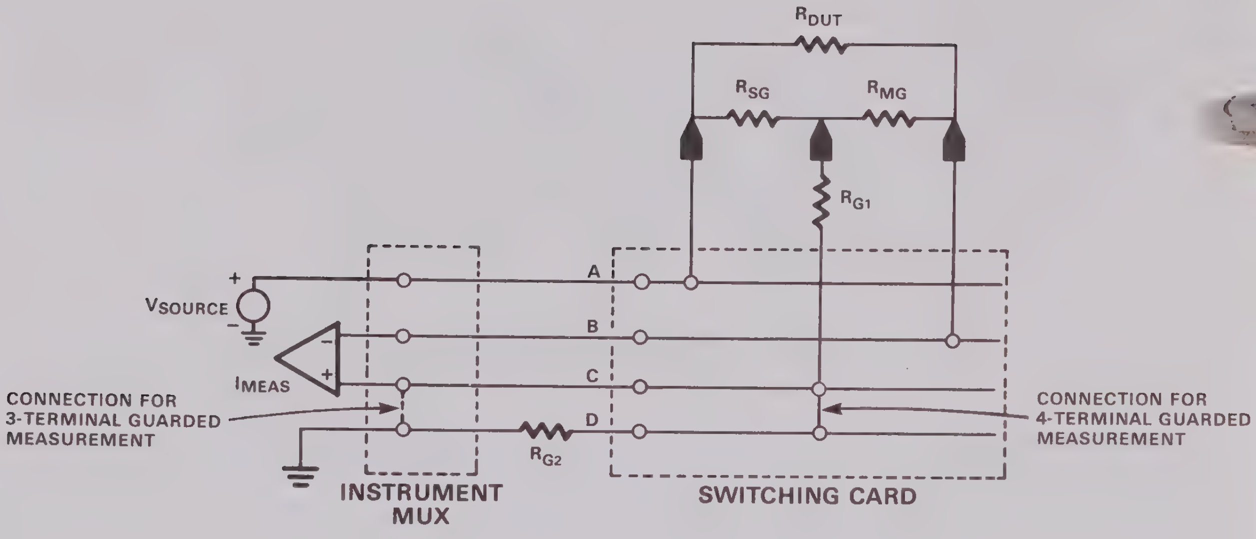


Figure 1-15. Guard Reference in Scanner Switching Card.

much greater than 10 (Rmg << Rdut), the inevitable slight voltage errors in the guarding circuit will produce currents through Rmg of the similar order of magnitude as those through Rdut, and the guarded measurement will not be very accurate.

When ATG writes a guarded test, it actually uses a technique that allows a more accurate measurement than the circuit in Figure 1-14 can produce. Instead of assuming that it can reference the current meter to the test nail's ground connection, it uses a fourth multiplexer line to connect the meter closer to the test nail than the ground point, as shown in Figure 1-14, so that the measurement is performed with 4 separate connections. Even though only 3 test nails are required (for 1 guarded node), the measurement circuit is for practical purposes a 4-terminal circuit, with separate connections for current return and guard sense. (More test nails are required if more components must be guarded.) The meter's ground input has a high impedance, so the guard current flows almost entirely through multiplexer line D (which is represented by Rg in Figure 1-15). Since virtually no current flows in line C, the meter's ground reference is much closer to the true potential on the guard node than it would be if the meter were connected to ground through the instrument mux (via the dashed line in Figure 1-15).

ATG writes an analog test for a device into the test program by evaluating the accuracy of various possible measurement configurations. Each time it tries to select a test, ATG evaluates all the possible source, measure and guard connections to a component, calculates the probable error, and finally accepts only a measurement configuration that yields the required accuracy. In some cases, ATG will select a simple, 2-terminal measurement; more often, however, it will select a 4-terminal circuit of the type shown in Figure 1-15. This is the best configuration that ATG can produce entirely on its own.

4-Terminal Kelvin and Guarded Measurements

ATG will not write tests requiring more than 1 test nail per guard node. However, once the user has acquired some familiarity with the test language, ATG's tests can be improved or new tests can be written to deal with unusual testing situations.

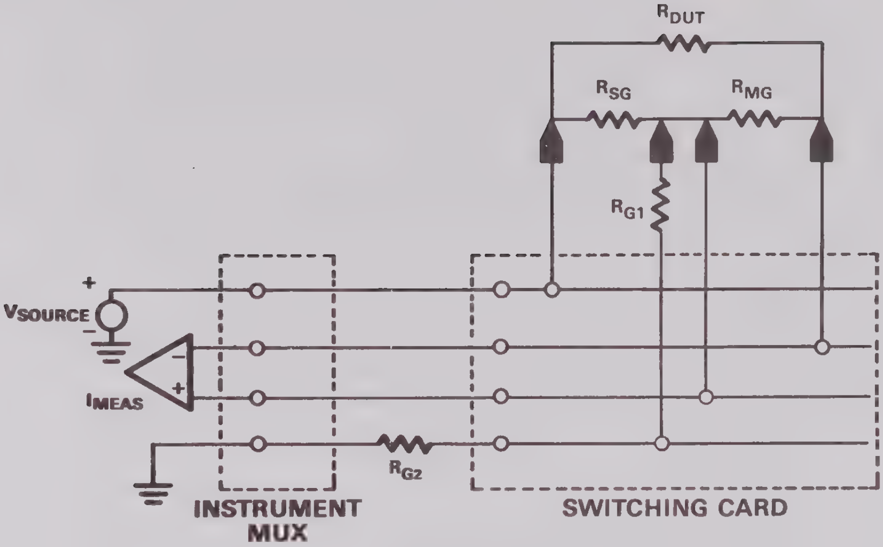


Figure 1-16. 4-Terminal Measurement with Guard Reference at Guard Node.

For example, if Rsg in Figure 1-15 is less than 5 to 10 ohms, enough voltage will be developed across the nail and connected wires, represented by Rg1, to introduce a significant guard error. To remove this source of error, the 4-terminal, 4-nail measurement scheme shown in Figure 1-16 could be used. Here the current meter's ground reference is brought directly to the node through a path carrying virtually no current, so that a smaller error voltage will be developed between the node and the meter. (Other significant errors, however, will arise in this situation. Careful analysis of the measurement circuit may show that voltages, in addition to current, should be measured to give an accurate result.)

Another kind of 4-terminal measurement is the Kelvin connection, shown in Figure 1-17. By measuring the voltage developed across the DUT with a high-impedance voltmeter, the voltage errors caused by currents in the measurement lines can be eliminated. This kind of test is very useful for measuring low-value

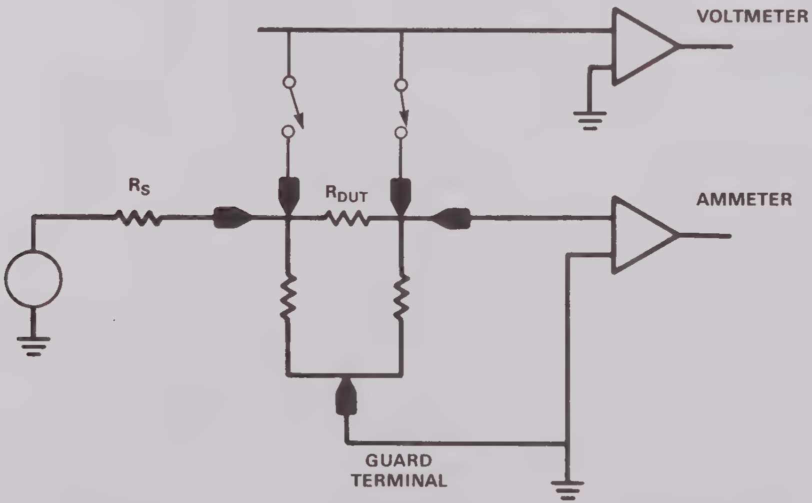


Figure 1-17. Kelvin Measurement with Guard.

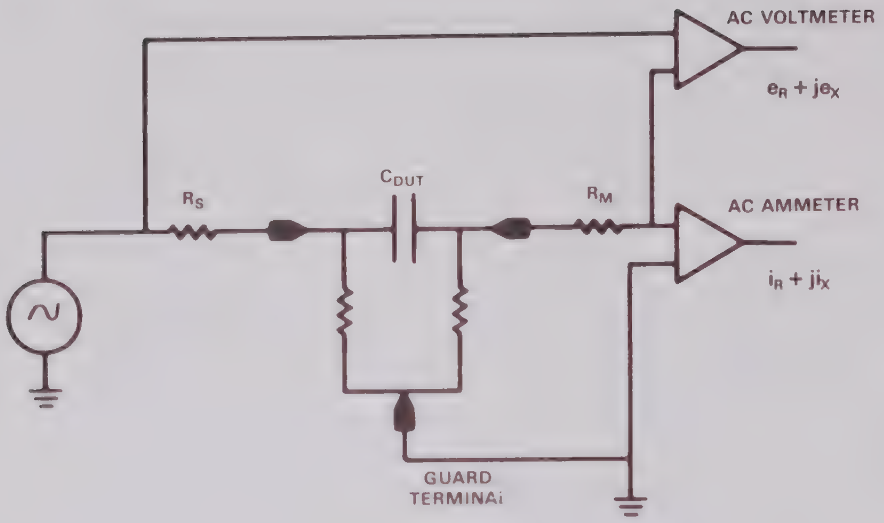


Figure 1-18. Typical 3-Terminal Reactance Measurement.

resistances (< 50 ohms). However, adding a guard to this circuit will require a fifth test nail, as shown in the figure. ATG will not write tests using Kelvin measurements; these must be programmed by the user.

Another source of measurement error is noise, or random variations in the value measured. When highly accurate measurements are required, these errors can be reduced by averaging 8 consecutive measurements. This technique is especially useful for accurate measurements of resistances greater than 1 megohm.

Refer to the 227X Test Programming manual for a more detailed discussion of analog test programming techniques.

Reactance Measurements

Reactance measurements are made in a similar fashion to the resistance measurements shown in Figures 1-11 through 1-16. Instead of dc measurements, ac measurements are performed at one of the 3 available frequencies, 100 Hz, 1 kHz, or 10 kHz. ATG choses the frequency that yields the highest measurement accuracy. In the measurement shown in Figure 1-18, the ac measurement unit (ACZ in Figure 1-9) produces measurements of in-phase and quadrature voltage or current (i.e., the voltages or currents at 0 and 90 degrees relative to the applied voltage or current). The capacitance (or inductance) is determined from the out-of-phase current. Using complex arithmetic, the software system can calculate values for any of the usual equivalent-circuit ac parameters listed below.

Function	Meaning
RS	Series Resistance (ohms)
CS	Series Capacitance (farads)
LS	Series Inductance (henries)
XS	Series Reactance (ohms)
RP	Parallel Resistance (ohms)
CP	Parallel Capacitance (farads)
LP	Parallel Inductance (henries)
XP	Parallel Reactance (ohms)
Z	Impedance Magnitude (ohms)
Q	Quality Factor
D	Dissipation Factor

In typical test measurements, CP is used instead of CS except for large capacitors, where the reactance is less than 25 ohms. LS is used for typical test measurements.

In addition, a dc bias can be applied during an ac measurement (for example, to bias an electrolytic capacitor or a diode).

A typical test for an inductor will specify a series inductance and a resistance (or Q). Therefore, the ACZ will perform the test by applying an ac voltage to the inductor and reporting values for the in-phase current and the quadrature (90 degree out-of-phase) current. From the quadrature value the system software will calculate the series inductance and from the in-phase value, the series resistance.

1.3.3 Complex Analog Device Tests

There are two categories of complex analog devices: those that can be broken down into several basic analog components that can be tested by analog in-circuit tests, and those that must use another technique requiring power on the pc board. A resistor pack is an example of the first category and an operational amplifier is an example of the second. Library routines are available for a number of standard devices that are tested with analog techniques, including SCRs, and the LM301 operational amplifier. Other library routines can be written by the user for devices that appear frequently, such as filters or transformers with specific, known characteristics. These routines are written in a special library language, ATSL, described in the Test Library Programming manual.

1.3.4 Digital Device Testing

As with analog testing, digital testing requires both a method of testing the device and a method of isolating it from the surrounding devices or from adjacent buses. The device test requires a set of known logic pulses, applied to the device's inputs, and a set of expected output states which the device should produce. These input and output bit patterns, called test vectors, usually reflect the truth table associated with the device. To allow the device to function, power and ground are made available to the device during the test.

A test of a digital circuit component usually requires large number of simultaneous input and output signal pulses. Accordingly, the test hardware for digital testing consists of a relatively large number of driver/sensor circuits. In the 2270, each digital or hybrid test nail is connected to a unique driver/sensor, so that, if necessary, all of the driver/sensors in the system could be active simultaneously. A test program is written by specifying the state of each test nail -- connected or disconnected, driving or sensing, high or low — for each step in the test. Test programs for other 227X systems are written in the same way, as if a separate driver/sensor were available at each test nail; the actual number of driver/sensors is, however, much smaller than the number of available nails. The system software arranges connections from the driver/sensors through a relay-based multiplexing system, so that all nails can be connected to driver/sensors. This means, of course, that a program cannot actually use all of the test nails at once; but test programs rarely need to use more than perhaps 100 nails at any one time.

The output of each driver normally functions as a programmable voltage source, generating highs and lows at voltages determined by the test program. In addition, the driver can be programmed to act as a current source to test for shorts in digital nodes. To test open-collector, open-emitter, and tri-state devices, the driver/sensor can switch in a pair of pull-up and pull-down resistors. A sensor is simply a voltage comparator that returns a logic 1 for all voltages above a certain programmed threshold, a logic 0 for voltages below a second programmed threshold, and a mid-state value for voltages between the 2 thresholds.

A digital test is conducted by applying to the device's inputs as many input combinations as the circuit connections will allow (inputs tied high or low obviously cannot be tested; inputs tied together cannot be tested separately). Using the truth table or an appropriate set of test vectors, the tester checks the outputs for the expected high and low states. Within certain limits the device's current and voltage parameters, as well as signal delays, can be specified in the test; in-circuit digital tests generally, however, are meant to verify the device's logical functions rather than its electrical parameters.

More complex digital devices often require 1 or more clock signals to control the propagation of signals in the device. The clock pins can be driven just as any other input, by applying alternate high and low logic levels from a driver nail. The test must be arranged so that this clocking sequence continues in strict order for the length of the test. Systems with the SpeedPlus test hardware option, however, contain a separate, programmable clock generator, which can provide clock pulses from special nails. The clock signals may be at higher frequencies than the test pulses, but they are coordinated with these in some regular way — for example, by driving and sensing after every second rising clock edge. In the SpeedPlus subsystem the clock generator thus replaces the alternating high-low pattern that must be explicitly programmed in a standard digital test.

Where a circuit board contains on-board clock generation circuits, these must normally be disconnected or disabled during IC testing, so that they do not interfere with the clock signal supplied from the test system. With the SpeedPlus option, however, the test system can either synchronize its clock to that of the on-board clock or simply synchronize the test steps -- without supplying a separate clock signal at all.

Each digital IC being tested may be connected to a number of other devices on the board. Its inputs will normally be driven high or low by these devices, since power must be made available to much or all of the board at once. To force the DUT's inputs to the desired states, regardless of the voltages on

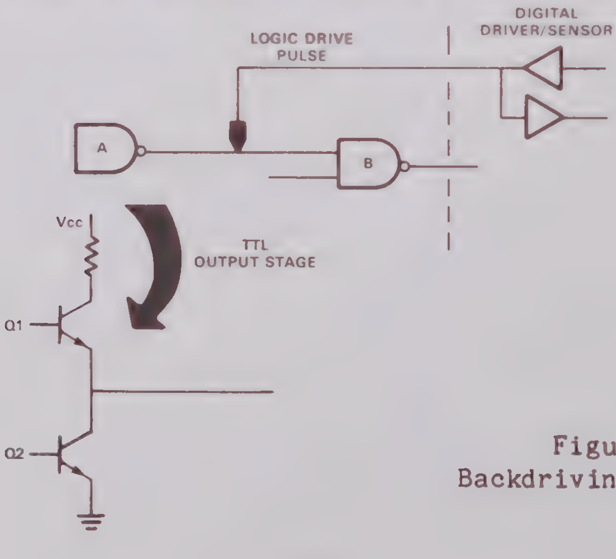


Figure 1-19.
Backdriving Digital ICs.

its input lines, logic pulses are applied with sufficient drive current capability to override the other devices connected to its inputs. This 'backdriving' technique is illustrated in Figure 1-19.

A low at the output of NAND-gate A indicates that transistor Q1 in the output state of the IC is conducting. In order to override this output, the tester must raise the output of the IC to a logic high, i.e., to about 3.5 V. It must therefore supply enough current to develop the required voltage across Q1.

The output resistance of the typical totem pole output stage shown here can range from around 20 ohms to 1 kilohm or more, depending on the output level. To override the output of such a totem pole the driver must supply more than 500 mA — a level that, in the worst case, could dissipate nearly a watt of power in the device's output. To prevent damage from high backdriving currents, the digital driver contains a current-limiting circuit that quickly reduces the drive current to around 40 mA after a logic state is generated. An important consequence of this safety mechanism is that a back-driven node cannot always be held at a logic level for more than a fraction of a second. (Approximate backdriving current versus time is shown in Figure 1-20.) The TTL backdriving mechanism operates in the same way, of course, for other logic families, as well.

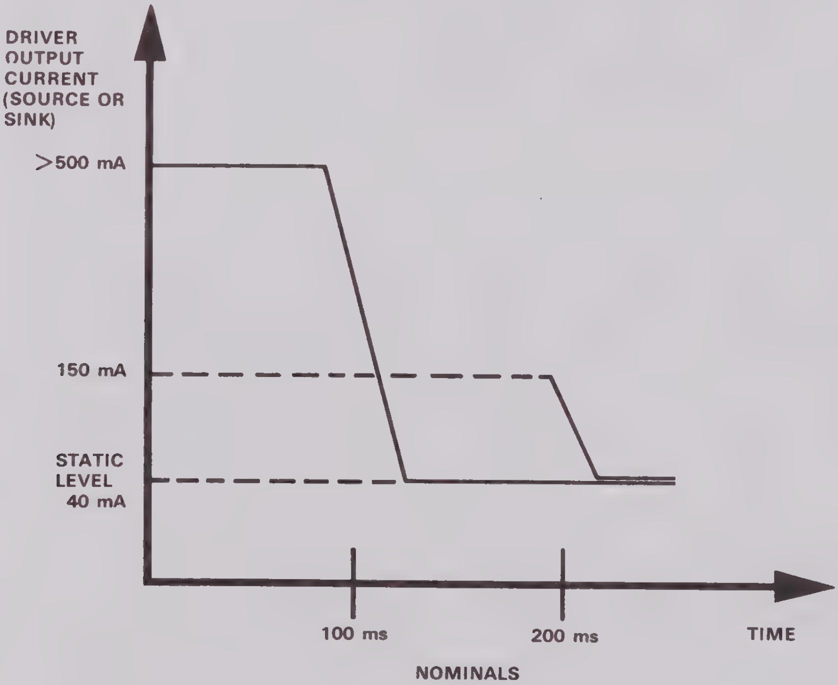


Figure 1-20. Driver/Sensor Drive Current vs Time.

Backdriving generally allows ATG to regard the inputs to the IC it is testing as isolated from the rest of the circuit, since the digital drivers will override the outputs of other ICs. However, it is often necessary to eliminate or reduce the effects of feedback loops which can create 'instability' at a node being controlled by a driver. Figure 1-21 shows how this problem can arise. In order to test the flip-flop, IC1, the tester drives nodes 100, 101, 102, 105, and 107 while monitoring nodes 103 and 104. The NAND-gate, IC2, can be affected by the activity of IC1 during the test. If node 106 is high, the NAND-gate output will change state whenever the flip-flop's Q output changes state. Now, although the tester is forcing node 100 from a low impedance driver, the driver is separated from the node by a length of signal wire, which behaves as a transmission line between the driver and the node. When the NAND-gate changes state the driver must respond instantly, but in fact its response is slowed by the intervening connection. Thus, the NAND-gate may be able to drive the node high or low for a few nanoseconds, until the driver current -- limited by the transmission line -- responds adequately.

Such instabilities are of short duration, usually less than 100 ns, and not of concern for most circuit configurations. However, even short noise pulses can affect controlling inputs of storage devices, such as the clock input of IC1 in Figure 1-21. At such inputs, backdriving alone is not always adequate to isolate the IC from interactions with other devices connected to it.

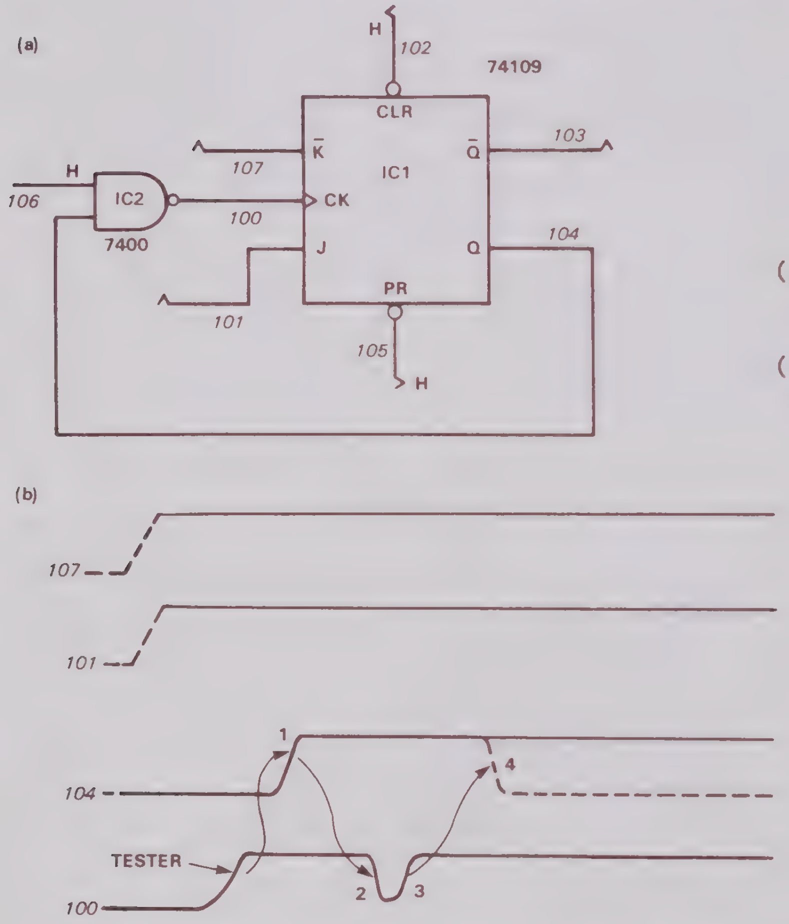


Figure 1-21.
(a) An example of a feedback loop inducing a noise glitch in a TTL circuit.
(b) Resulting waveform when testing the J-K flip-flop, IC1, in toggle mode. The tester causes transition (1) by clocking a high at the J-input and a high at the K-input. Since node 106 is high, the output of gate 2 goes low momentarily (2). Backdriving forces gate IC2 high again (3). The flip-flop could toggle again (4), depending on the amplitude of the pulse.

Where possible, ATG consults the digital libraries to find out how to set the outputs of devices connected to DUT inputs to a high-impedance or tri-state condition, so that they cannot interfere with the test. Where this is not possible, it sets them to an 'inhibited' state, in which the outputs remain active but will remain stable while the DUT is being tested. This helps prevent 'glitches' that can be caused by logic activity on the board feeding back from the outputs of the DUT which is being stimulated.

To be practical, digital tests must be conducted at a rate much higher than they could be executed by the tester's central processor. This is accomplished by using the processor to load the test vectors from the test program into a bank of memory cells attached directly to the driver/sensors; a special controller running at high speed then reads out the test vectors from this memory directly to the drivers and reads in the test results, i.e., the sensor inputs. The test pattern loaded into this local driver/sensor memory is called a BURST; the size of the burst is limited to about 460 test vectors by the size of this memory. (Longer tests are performed by breaking them into several bursts.) After a burst is complete, the central processor moves the data it requires back into main memory for analysis. The standard digital test hardware can execute test steps at up to about 280 thousand test vectors per second; tests conducted on the SpeedPlus optional hardware can execute at up to 5 million test vectors per second.

VLSI devices present a special class of problems for digital testing. Many VLSI devices are so complex that it is impractical to test their truth tables. Instead, the device is initialized (e.g., by activating its reset pin), and the test system applies a relatively small but carefully chosen sequence of inputs (commands or instructions). It monitors the outputs (data and address lines, control lines, etc) for the correct responses. The clock pin (or pins) must be driven continuously, by driving the associated test nail with an alternating pattern of highs and lows in the test vectors. In some cases (while waiting for a reset to complete, for example), the device's intermediate activity does not need to be monitored, and the test program can be made shorter by having it wait in a loop until the anticipated result appears.

Tests for a large number of digital components exist in the Digital Library. These tests are automatically selected by the ATG task. A list of these components is given in Appendix C.

1.3.5 Digital Bus Testing

The sequence of digital testing is organized so that non-bused ICs are tested first. Then, before full tests are performed on the bused devices, the bus connections are tested by attempting to free the bus from all the devices connected to it. A method for setting each bused device's outputs to a high impedance state (called the disable procedure) is provided in the Digital Test Library (typically the test must drive a control pin such as "chip enable"). ATG extracts the relevant procedures for all ICs on the bus. Then the driver/sensors test the bus by attempting to drive the bus lines high and low from high-impedance voltage sources.

Each driver/sensor contains built-in, programmable pull-up and pull-down resistors that connect to the logic high and low drive voltages. These resistors are commonly used to simplify the sensing of open-emitter and open-collector outputs. In bus testing, the pull-down resistors are first used to drive the bus lines low. If the pull-down resistors can bring the bus lines to a low voltage level, it can be assumed that the bus is in a high impedance state, at least in the high voltage range. Then the programmable pull-up resistors are connected to the logic high voltage, and the bus lines are checked for logic highs. This verifies that no device is driving the bus at the low logic level. Bus tests are performed in the same fashion at nails connected to the optional SpeedPlus hardware.

A pull-up or pull-down resistor driving a high-impedance load (the tri-stated bus) will normally produce a voltage near the high or low sense thresholds. However, if a device on the bus is stuck in a high or a low state, it will overcome the resistive source and produce a logic high or low, so that one of the expected bus logic values will not be sensed. The bus test can determine the failing logic level, but it cannot tell which device is driving the bus to that level, since all of the IC outputs are connected together. The

IC causing a bus fault must still be identified and replaced before other ICs on the bus are tested, since the tester can only perform a valid functional test of a bused IC's outputs if all other ICs on the bus are effectively disconnected from the bus. Whenever a faulty bus line is found, the bus test fails and the test proceeds to invoke the BUSBUST diagnostic technique to isolate the IC causing the bus failure.

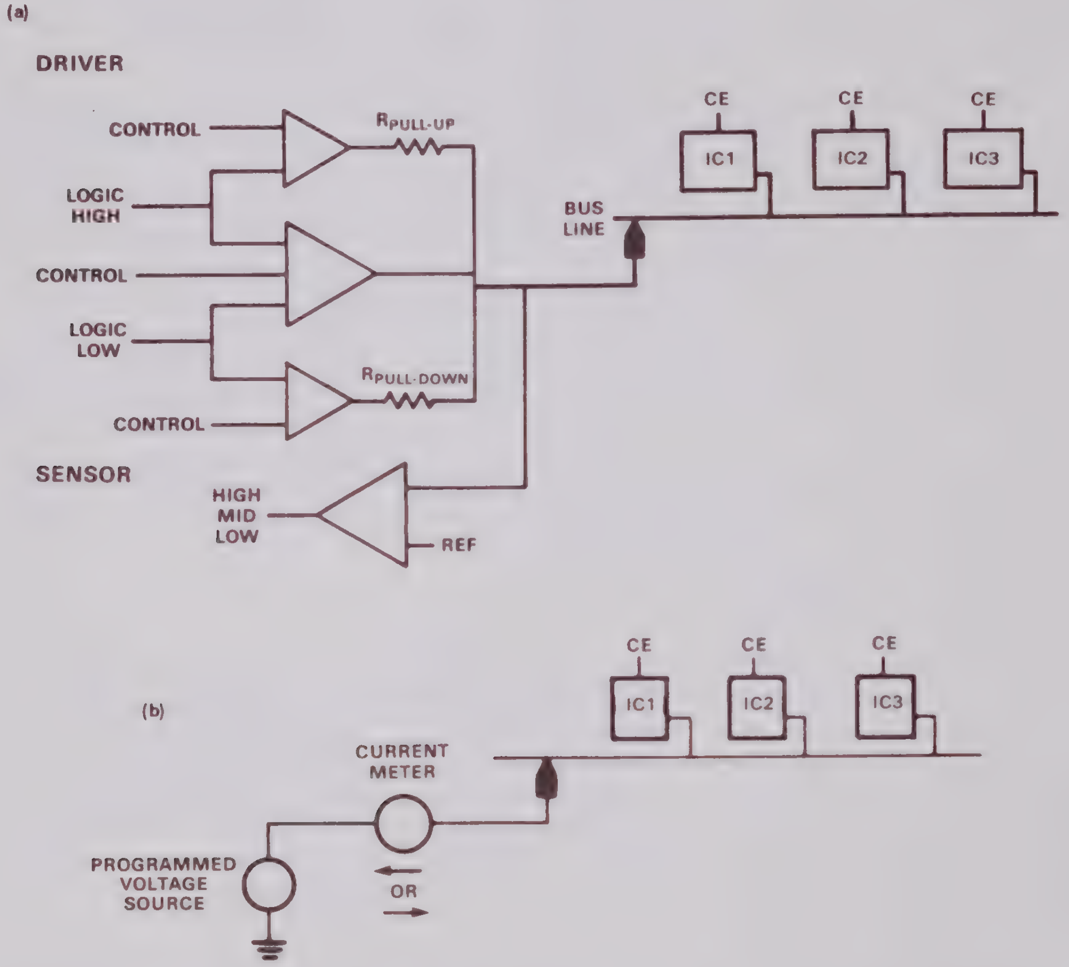

(a)
DRIVER
CONTROL
$R_{PULL-UP}$
LOGIC HIGH
CONTROL
LOGIC LOW
$R_{PULL-DOWN}$
CONTROL
SENSOR
HIGH
MID
LOW
REF
BUS LINE
CE
IC1
CE
IC2
CE
IC3
(b)
CE
IC1
CE
IC2
CE
IC3
CURRENT METER
PROGRAMMED VOLTAGE SOURCE
OR

Figure 1-22. Measurement Configuration for Diagnosing Bus Faults.
(a) While Testing Tri-Stateable Buses.
(b) While Seeking Faulty Bused Device.

This diagnostic technique involves making accurate current measurements while ICs on the bus are selectively and methodically enabled to cause bus activity, as shown in Figure 1-22. As was explained, a bus fault arises because some IC on the bus does not disable all of its outputs when commanded to do so. Instead, it continues to drive one of the bused lines high or low. To determine which IC is still driving the bus, the tester now draws additional information about all ICs on the bus from the system library. Using this information, it commands each IC in turn to drive all the bus lines to the same state, high or low, as the faulty line. All the other ICs remain disabled. In most cases the tester will find that the active IC causes an increase in the current into or out of the faulty bus line; if the command is issued to a defective IC, however, the current will remain the same. The IC that causes no current change in the faulty line is then identified as defective.

If no bus faults are found, each IC connected to the bus can be tested for correct functioning. All other ICs on the bus are disabled, as in the bus test, and the system then performs the same sort of functional test used for non-bused ICs. Since the bus test has determined that the system can successfully free the bus from the other ICs, the IC test does not need to concern itself with bus interactions; again, the IC test is conducted as if each IC were the only device on the board.

1.3.6 Testing VLSI Devices

As the complexity of a digital IC grows, it becomes increasingly difficult to test it with the same methods used for functional testing of smaller ICs. For one thing, the sheer size of the device's truth table makes a complete functional test impractical. Full functional verification of a microprocessor, for example, could require from 10,000 to many millions of test vectors (input/output patterns). Instead, methods must be developed that do not blindly test all possible combinations of inputs and outputs, but rather verify individual functional units within the IC (such as I/O registers, ALUs, control sections) and the data paths among them. This approach, it turns out, requires far fewer test vectors, since the truth tables of the functional units are often relatively small. Provided that at least some general information is available about the internal structure of the device, this approach can produce highly effective tests using a practical number of test vectors.

Speed requirements are a second problem that VLSI devices often present. Devices that use dynamic storage usually will not function correctly below a certain minimum clock rate, and this clock rate in turn imposes a minimum test rate. When this minimum rate exceeds the 270-kHz maximum test rate of the standard hardware, an alternating high-low pattern in the test vectors (as described in para 1.3.4) can no longer be used to supply the required clock input. Instead, special clocking hardware must be installed in the test fixture, or the UUT's own clock must be enabled during the test. Some method must then be found to synchronize the slower test steps with the device. Typically, the test program will use a WAIT or HALT pin on a processor to keep the processor from advancing from one instruction to the next any faster than the test system permits. In some cases, the test program can enter a loop, testing and waiting until a desired condition (address, data, or machine state) is sensed.

A third problem in testing many VLSI devices is that meaningful outputs do not always immediately follow their inputs. A numerical processor, for example, may require hundreds of clock cycles to produce a floating-point product or transcendental function. Larger microprocessors present a similar problem because they frequently execute machine cycles in which nothing of interest to the test program occurs. During these 'dead' cycles the test system must continue to supply test vectors in order to stay in synchronism with the processor. Consequently, many test vectors can be wasted in keeping the test program in step with the UUT, and the number of meaningful input and output patterns included in the test program may be severely limited.

While it is possible to write good tests of most VLSI devices using the standard 227X digital hardware, an optional digital test subsystem is available on the 2271 and 2272 systems to solve the specific problems of testing VLSI devices. Called the SpeedPlus option, this subsystem provides a number of additional test nails that can be used either as standard test nails with conventional test programs or as special 'high-speed' nails for executing in-circuit tests of VLSI devices. To store the larger test programs that these devices often require, additional memory is provided for test vectors. In addition, the SpeedPlus subsystem incorporates special-purpose hardware to furnish clock signals and to permit 'stop-and-go' synchronization of the test program to the UUT.

With these special testing facilities, the subsytem can turn the difficulties imposed by VLSI devices to advantage. Because a typical VLSI device's clock input(s) require only a regular, steady signal, the test program actually needs only to specify this signal once and leave the clock generator to run. This saves executing a test vector every time the clock input must change state. Since many intermediate device states may not need testing (while calculation or instruction decoding are under way, for example), the test program can afford to stop during these intervals and continue only when relevant states occur. A SpeedPlus triggering facility permits each test step to wait for some defined condition. This can be a machine state, a bus cycle, an address, or indeed any condition that can be recognized as a combination of nail states. This saves additional test vectors that would have to be executed merely to keep the tester in synchronization with the UUT.

For testing VLSI devices, the SpeedPlus option provides:

test rates of more than 20 times the standard 227X speed

storage for up to 7 thousand test vectors

built-in, programmable clocking hardware

programmable synchronization hardware

While the SpeedPlus hardware does allow very precise control of many timing parameters used in a test, it should be pointed out that in-circuit testing does not generally allow valid tests of device timing. The artificial environment imposed by the in-circuit test -- the disabling of adjacent circuit-board components and substitution of the test system's driving and sensing impedances at the nodes being tested -- means that any timing measurements made during the test, however accurate, will have little relation to the device's timing characteristics when it operates in its true circuit environment. Where timing parameters are critical to a circuit's operation, only a functional test of the board or a board sub-section is likely to give meaningful results.

1.4 FUNCTIONAL TESTING

Functional testing can be provided for with the inclusion of such options as additional scanner modules, additional power supplies, and the IEEE general-purpose instrumentation bus. These allow the user unlimited flexibility in designing instrumentation and programs for functional tests. The bed-of-nails interface allows any combination of stimulus and measure nodes, from edge connector tests to single device tests. Software for functional tests must be written by the user but, once it is prepared, it may be automatically inserted into the test program via an Automatic Test Options (ATO) file. This feature of the test system permits hand-written tests and modifications to be added to a test program automatically, each time it is compiled, so that a functional test for a board can be developed and debugged independently of the in-circuit test, much as individual component tests are developed. If the circuit board is later modified, but its functions remain the same, the functional test can be included automatically in the new test progam that ATG writes for the new circuit description.

With simpler circuit boards, a fast and efficient test strategy can sometimes be realized by writing a thorough functional test for the board which executes as the first part of the test program. The in-circuit test then executes only if the functional test fails; it identifies the cause of the failure by identifying the faulty component and records failure statistics for production control. Because the functional test can execute faster than the in-circuit test, good boards are processed more quickly than they would be with an in-circuit test alone. At the same time, the in-circuit test gives a relatively quick, detailed analysis of those faults that do arise, to speed repair and adjustment of production parameters.

A thorough test for a larger board can follow the opposite strategy. First, all of the components are tested individually, using the test written by ATG from a circuit description. The test engineer can then add a statement to each in-circuit component test to record whether or not it passes. If all components pass their in-circuit tests, the test program can then proceed to a short, manually-written section that applies power to the circuit board, feeds it some appropriate input signals (a set of logic vectors or analog signals), and tests whether the board produces its specified outputs. This functional test does not need to be exhaustive, since by this time it is known that all of the components are performing correctly. Instead, it should try to check basic board-level functions (does the board reset and enter its start-up routine?) and critical, cumulative parameters (does it supply correct voltages and currents to its outputs, is a signal-to-noise ratio within limits?), as well as any peculiar failure modes that have revealed themselves in the course of developing and testing the board.

A sophisticated functional test can furnish more detailed diagnostic information by performing further functional tests of board sections and components. Because the in-circuit test fixture provides access to the board's internal nodes, the test programmer can write further tests into the program that localize the functional problem to a specific section or component on the board. Starting from a particular board-level functional problem, a program section can be written that descends a trouble-shooting tree, performing successively more localized tests until the fault can be identified. Because the in-circuit test has already established that all components and connections are working, the functional test will be directed only to specific critical parameters (for example, an op-amp's input noise or slew rate, or a clock crystal frequency) not covered by the in-circuit test.

System Monitor – Section 2

2.1 INTRODUCTION

2.1.1 General

The GenRad Board Test Workstation monitor is the user interface to all the system software provided for performing program preparation and pc board testing. It consists of a monitor page for each major mode which leads the user to all the selectable options for performing the desired task. There are also many utility pages that are used for the manipulation of files. Figure 2-1 is a diagram showing the levels of software that interface with the various monitor modes. Table 2-1 lists all the major mode names and utility names and the task that each page performs.

The descriptions of monitor pages and the reasons for selecting each option are given in the sections that discuss the function.

2.1.2 Brief Description of Monitor

Since the monitor is used for all normal operations of the test system, it is automatically put into operation when the test system is initialized. The Diagnose options page is the page needed for testing boards, so it appears on the display at initialization.

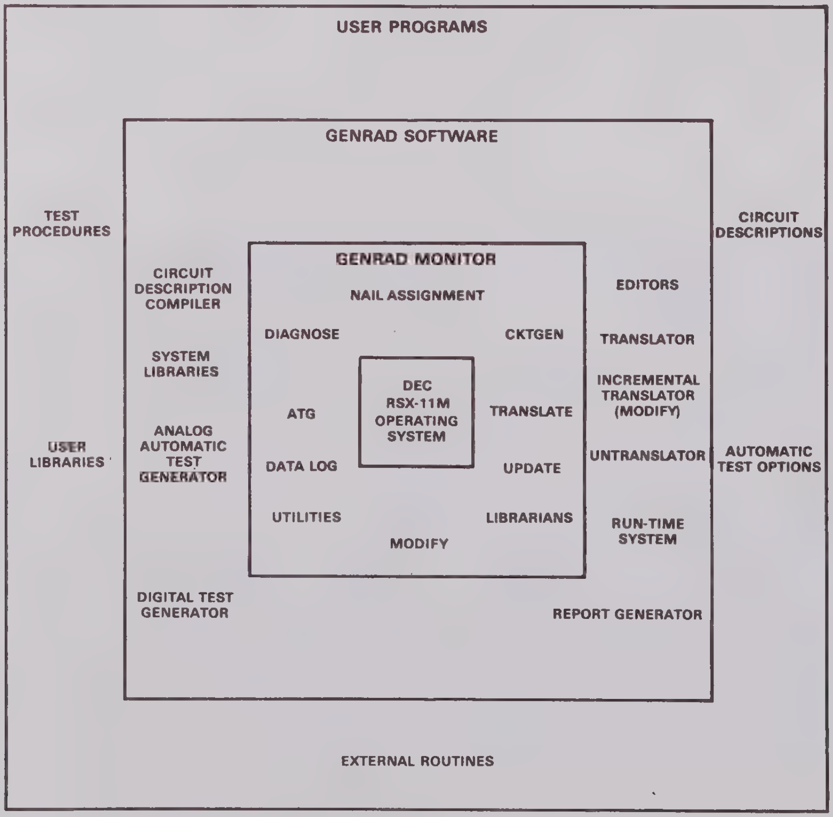

USER PROGRAMS
TEST PROCEDURES
USER LIBRARIES
CIRCUIT DESCRIPTIONS
AUTOMATIC TEST OPTIONS
EXTERNAL ROUTINES
GENRAD SOFTWARE
CIRCUIT DESCRIPTION COMPILER
SYSTEM LIBRARIES
ANALOG AUTOMATIC TEST GENERATOR
DIGITAL TEST GENERATOR
EDITORS
TRANSLATOR
INCREMENTAL TRANSLATOR (MODIFY)
UNTRANSLATOR
RUN-TIME SYSTEM
REPORT GENERATOR
GENRAD MONITOR
NAIL ASSIGNMENT
DIAGNOSE
CKTGEN
ATG
TRANSLATE
DATA LOG
UPDATE
UTILITIES
LIBRARIANS
MODIFY
DEC RSX-11M OPERATING SYSTEM

Figure 2-1. Levels of Software.

Table 2-1
MAJOR MODES AND UTILITIES

Major Mode Name	Function	Para
ACL	Calls Analog Component Librarian	Library[1]
ATG	Generates test program from circuit description	7.9
ATL	Calls Analog Test Librarian	Library
CKTGEN	Generates circuit description from components list	6.3
DIAGNOSE/LIMIT (TEST Sub-mode)	Tests pc boards	Operate[2]
DIAGNOSE/UNLIMIT (DEBUG Sub-mode)	Debugs test programs and fixtures	9.4
DTL	Calls Digital Test Librarian	Library
FIXTURE REPORT	Generates nail assignment reports for 2270	8.7
LOG	Displays data collected during board testing	Operate
MODIFY	Allows incremental changes to binary test file	9.6
NAIL ASSIGNMENT	Assigns nails, produces nail and fixture reports for 2271/72/75	8.6
TRANSLATE	Translates test program into binary form	9.2
UPDATE	Matches .TPG to .OBC file using .DBT file	9.5

Utility Name	Function	Para
BATCH	Runs system under batch control	Operate
COPY	Copies contents of file(s) to another file	3.5.1
DCHANGE	Allows disk changes without reinitializing system	3.5.2
DCOPY	Copies all files from one disk to another	3.5.3
DELETE	Deletes files from disk device	3.5.4
DFORMAT	Formats new disks to ready them for use	3.5.5
DIRECTORY	Lists file directory of disk device	3.5.6
EDITOR	Runs text editor to create and revise files	11.2
FREE	Shows amount of unused space left on a disk device	3.5.7
LEDITOR	Runs read-only editor for viewing text files	3.5.8
NETWORK	Transfers files to/from remote system using GRnet	Operate
PRINT	Lists file to printer	3.5.9
PURGE	Removes old versions of files from disks	3.5.10
RENAME	Changes file name without creating new file	3.5.11
SETUP	Provides library names, reports system configuration	4.9
SPLIT/MERGE	Transfers file to dissimilar disk device	3.5.12
TYPE	Lists files to video display in screen-sized pages	3.5.13
UNLOCK	Unlocks improperly closed RSX-11M files	3.5.14

---

[1] 227X Test Library Programming manual.
[2] Operating manual for applicable system.

Monitor Access Levels

The monitor is designed to be run for two basic purposes: testing boards and preparing programs. When being used for testing boards, it can be switched to a restricted condition where only the options used in testing are accessible to the user. This is the limited access level of monitor operation. When any task other than testing (or initiating certain batch jobs) is to be performed, the authorized user would switch into the unlimited access level. The entire monitor is then available. Switching between monitor access levels is only done by authorized users who have a valid password.

Edit and Command Modes

There are two classes of actions that can be taken when a monitor options page is displayed: editing the page options (screen-editing mode) and issuing commands (command mode). The page is first displayed in command mode as denoted by the # symbol (prompt) at the bottom left of the page. Characters typed at the keyboard will appear to the right of the prompt and are read by the monitor as a command (assuming a legal command is typed). If, however, the character typed is a special page option editing command, the display will automatically revert to screen-editing mode and editing instructions can be typed until it is desired to return to command mode. The screen-editing mode provides a convenient way of specifying page options, but page options may also be specified from command mode.

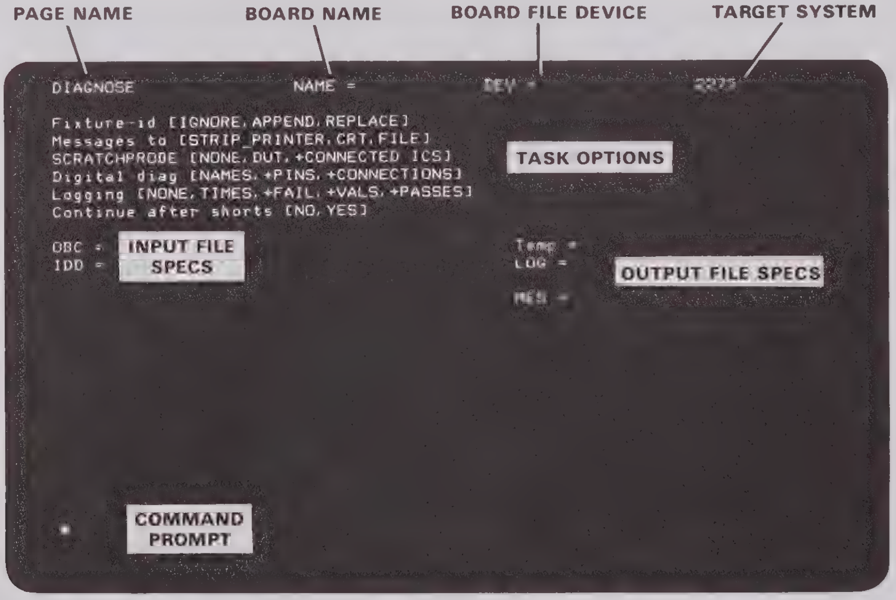
The entire name is not required when calling a major mode or utility; only enough letters are needed to distinguish it from other names. For example, since BATCH is the only name beginning with the letter B, the single letter B is sufficient to call the Batch utility. To call the Directory utility, however, 3 letters are needed to distinguish it from the Diagnose mode.

Monitor Page

All monitor pages are organized as illustrated by the Limited Diagnose page in Figure 2-2.

Specifying Options

The screen-editing mode of specifying options on a monitor page involves knowing only a few simple editing commands. All that is required to enable an option selection is to place the cursor on the desired selection using the screen editing commands; then leave the option by moving to another option or returning to command mode. For example, to move the cursor from the prompt to



```
DIAGNOSE                NAME =                DEV =               2273

Fixture-id [IGNORE,APPEND,REPLACE]
Messages to [STRIP_PRINTER,CRT,FILE]
SCRATCHPRODE [NONE,DUT,+CONNECTED ICS]
Digital diag [NAMES,+PINS,+CONNECTIONS]
Logging [NONE,TIMES,+FAIL,+VALS,+PASSES]
Continue after shorts [NO,YES]

OBC =                                         Temp =
IDD =                                         LOG =

                                              MES =
```

Figure 2-2.
Monitor Page Layout Illustrated by Limited Monitor Diagnose Page.

the "NAME =" option field at the top of the page, press the CTRL key and the T key simultaneously. Now the name of the circuit board test program can be typed in and is thus specified.

To specify the device of the test program, press the RETURN key to move cursor to the next option field and type in the device name. Pressing the RETURN key again will move the cursor to the first choice in the "Fixture-id" option, namely, IGNORE. If the REPLACE option is desired, press the spacebar twice to move the cursor twice in the selection field. The REPLACE option is thus selected and will be displayed in reverse video. The filling out of the options page can be continued in the same way. When the cursor is at the last option field and the RETURN key is pressed, the prompt will re-appear and the diagnose task can be executed by pressing the RUN key.

With the descriptions presented so far in this section, a new user can start using the monitor. The remainder of the section describes in more detail the many additional features of the monitor.

2.1.3 Monitor Organization

The monitor consists of a top-level command mode from which major modes and utilities can be called and run. Each major mode and utility is a separate mode and has its own monitor options page.

Options pages have the standardized format shown in Figure 2-2. To ease and speed the use of the monitor, the options on every page appear in one of three forms: yes/no choices, multiple choice selections, and fill-in type responses. Note that yes/no options are presented in multiple choice style for consistency of appearance and function on the page. The number of questions on the page depends on which sub-mode and options the user chooses. Only those options relevant to the mode (based on the setting of other options) will be presented to the user. For example, in Diagnose mode, if the "Logging" option is set to NONE, the "LOG =" file specification line does not appear on the page.

The left-most selection in multiple choice option fields is the default setting. For example, the Diagnose options page "Fixture-id" default setting is IGNORE. On the standard video terminals, the default settings appear underlined when enabled; other option settings are displayed in reverse video when they are selected. On hardcopy terminals, the enabled setting is displayed in capital letters and other selections appear in lower-case letters. On the optional VT10X terminal, all selected options are underlined if the cursor option is set to "blinking underline" or are displayed in reverse video if the cursor option is set to "blinking block." Blinking block is the normal VT10X mode.

The "NAME =" and "DEV =" option values are monitor-wide. Although selections and entries are made on any monitor options page, they are displayed on all pages. Entries made on one options page are automatically filled-in on all pages where they are used.

Many options pages provide for the selection of one of several modes of operation. The mode is selected by the mode option, which is the first option line on the page and is a multiple choice option. In these cases, the options displayed are those which apply to the selected mode. Therefore, an options page may have several forms. Figure 2-3 shows the system monitor structure of monitor pages with their operating modes.

The monitor operates from either of two access levels, LIMITed or UNLIMITed. The access level is controlled by the monitor's LIMIT and UNLIMIT commands and may be changed by a qualified user at any time by issuing the appropriate command and responding with the proper password. The two levels provide a means of protecting software during the "everyday testing" operation and of shortening the setting-up procedure. LIMITed access restricts a user to loading and executing previously prepared test programs and running batch jobs. The running of modes in a batch job is not limited. Enabling UNLIMITed access provides the user with all the capability of LIMITed access, as well as the capability to create and debug test programs, to display test results, and to manage disk file storage.

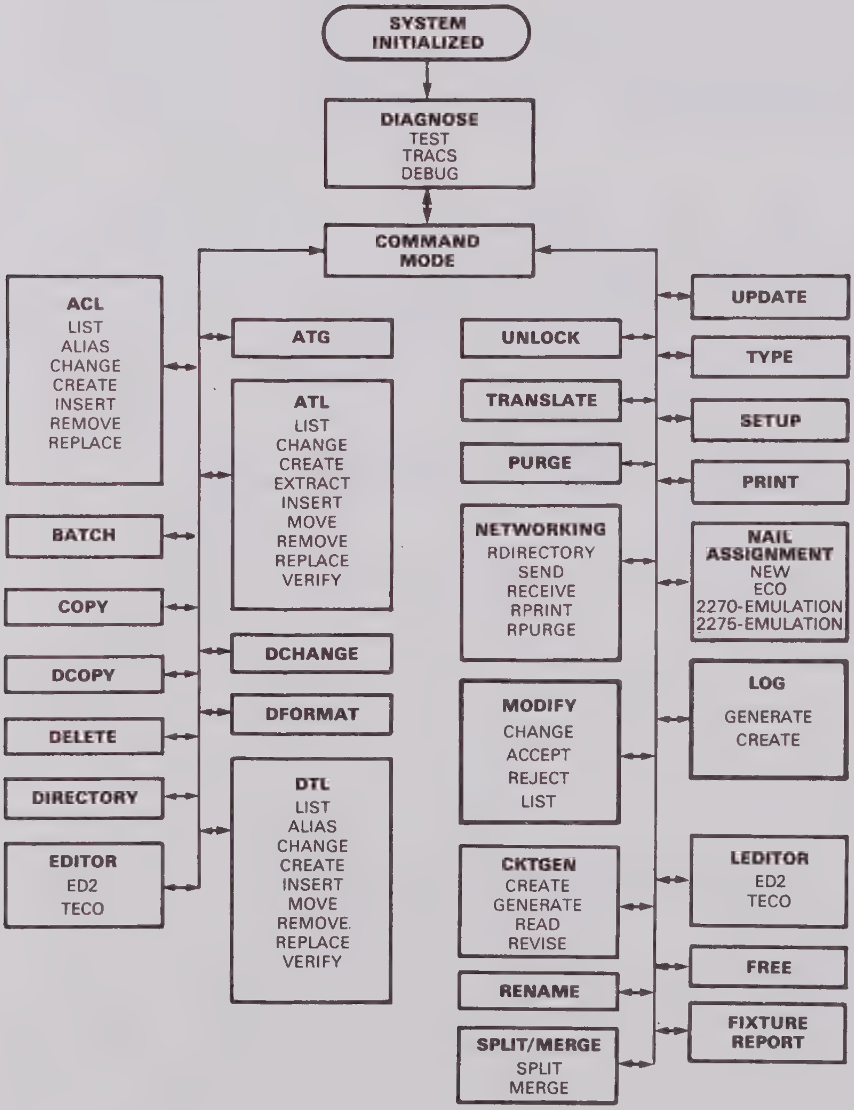

SYSTEM INITIALIZED
DIAGNOSE
TEST
TRACS
DEBUG
COMMAND MODE
ACL
LIST
ALIAS
CHANGE
CREATE
INSERT
REMOVE
REPLACE
ATG
ATL
LIST
CHANGE
CREATE
EXTRACT
INSERT
MOVE
REMOVE
REPLACE
VERIFY
BATCH
COPY
DCOPY
DELETE
DIRECTORY
EDITOR
ED2
TECO
DCHANGE
DFORMAT
DTL
LIST
ALIAS
CHANGE
CREATE
INSERT
MOVE
REMOVE
REPLACE
VERIFY
UNLOCK
TRANSLATE
PURGE
NETWORKING
RDIRECTORY
SEND
RECEIVE
RPRINT
RPURGE
MODIFY
CHANGE
ACCEPT
REJECT
LIST
CKTGEN
CREATE
GENERATE
READ
REVISE
RENAME
SPLIT/MERGE
SPLIT
MERGE
UPDATE
TYPE
SETUP
PRINT
NAIL ASSIGNMENT
NEW
ECO
2270-EMULATION
2275-EMULATION
LOG
GENERATE
CREATE
LEDITOR
ED2
TECO
FREE
FIXTURE REPORT

Figure 2-3. System Monitor Control Page Structure
(showing each major page with all the sub pages).

2.2 MONITOR OPERATION

2.2.1 General

The system monitor is enabled by the start-up procedure presented in the appropriate operation manual. The start-up procedure initializes the test system and, when initialization is completed, displays the Diagnose mode options page.

The system monitor is used for disk file management, test program preparation, board testing, and test data logging. The functions are implemented as tasks and utilities with menu-type option pages. Methods of selecting and enabling the options are described in this paragraph.

Monitor control is primarily exercised from the keyboard or terminal. The keypad is used to perform two primary functions: running previously prepared test programs and starting batch jobs. All other tasks, such as using the editor, digital librarian, automatic test generator, translator, etc, require a keyboard or VT10X terminal. (The VT10X terminal is available only on multi-user Foreground/Background systems.)

2.2.2 Command Mode

The command mode is the top level of the system monitor. When the test system is initialized the Diagnose options page appears and the command mode is enabled. The presence of the command mode on any options page is indicated by a sharp sign prompt (#) in the lower left of the page. From this command line prompt the user can enter top-level monitor commands, change-page commands, command line directives, and single-line commands.

Throughout this description, examples of commands entered by the user are shown in the following format.

Format

```
# command
```

where:	#	Monitor prompt. Commands entered from the keyboard are displayed at the prompt. In the lower-level monitor display the prompts TEST]> or DEBUG]> are used.
	command	Any of the monitor commands.

The following keys on the keypad, keyboard, and terminal are functionally equivalent:

Function	Keypad Key	Keyboard Key	VT10X Terminal Key
Enter a command:	ENTER	RETURN	RETURN
"RUN" key:	START/CONTINUE	RUN	LINEFEED

Top-Level Commands

Top-level monitor commands are entered from the command line on any monitor options page. Table 2-2 lists the top-level commands and their functions.

Table 2-2

TOP-LEVEL MONITOR COMMANDS

Command	Function
ABO or ABORT	Exits from the GenRad monitor and returns system control to the RSX-11M operating system. (Same as EXIT, SHUTDOWN.)
EXI or EXIT	Exits from the GenRad monitor and returns system control to the RSX-11M operating system. (Same as ABORT, SHUTDOWN.)
FLIMIT	Enters the limited access level without requesting a password. Once this command is issued, unlimited access can not be re-enabled without a password.
GRTRM	Sets the terminal display characteristics to those required by the keyboard and video display (normally preset to correct value). Option selections appear underlined or in reverse video to indicate the enabled setting. Screen editing can be used from GRTRM. (Related commands: HCTRM, VTTRM.)
HCTRM	Sets the terminal display characteristics to those required by a hardcopy terminal and rewrites the options page. The hardcopy display shows option selections in mixed upper and lower case, with the enabled setting in upper case. Screen editing cannot be used to select options from HCTRM. Any hardcopy or video terminal capable of printing at least 80 characters per line may be used. (Related commands: GRTRM, VTTRM.)

HELP	Displays helpful information. The first time HELP is called, a description of the current monitor page appears, along with instructions for obtaining more system help via a series of descriptive pages. When the series is completed (or the user aborts the HELP series), the monitor options page from which HELP was called is returned to the screen. An option name can be included in the command to get help on a specific option or file spec. For example, if the "Logging" option is set to NONE, entering
	`#HELP LOG`
	displays the help message:
	`Data logging is disabled`
	and the screen-edit mode (in the LOG option) is enabled.

LIMIT	Enables the monitor's LIMITed access level, allowing only the Test sub-mode of Diagnose mode and batch jobs to be run. Entering
	`#LIMIT`
	causes the system to request
	`Password?`

Table 2-2 (continued)

Command	Function

LIMIT (cont)	If the password is incorrect, the command is ignored. Entering the proper password enables the LIMITed access level and gives the user an opportunity to change the access password with the request
	`New password (<RETURN> to keep the same):`
	Pressing RETURN leaves the password unchanged. When the system is first received, the initial password is GENRAD. User-created passwords can be up to 32 alphanumeric characters in length. If a new password is entered, the system verifies the entry by requesting
	`Retype new password for verification:`
	A verification failure will be reported; otherwise, the new password replaces the old one. The access password is necessary to return to unlimited monitor access with the UNLIMIT command.

PAGE	Re-writes the currently displayed monitor options page. Useful when a system message disturbs the display.
PAUSE	Used in a batch file to cause execution of the batch file to wait for user response. The command caused the message "press space bar to continue" to be displayed.
RESTORE	Restores all monitor options and file specs on all pages to the states they were in when the SAVEALL command was issued or at system startup. If the SAVEALL command included a file name, the file name must also be included with this command.
SAVE	Saves the currently displayed monitor options in a batch input file. If no name is specified, the name defaults to DEV:NAME.BCH taken from the top of that monitor page. To specify an alternate name, include the name, separated from the command by a space character, on the command line. For example, `#SAVE BD` saves the current page as a batch file with the name DEV:BD.BCH. The device and standard batch file extension, .BCH, is automatically assigned. Running a batch file created with SAVE causes the monitor page to appear and options and file specs will be set as they were when the page was saved. To run the mode, the user must enter the RUN command on the displayed page.
SAVEALL	Saves all monitor options and file specifications from all pages in the monitor. A file name can be specified in the same manner as for the SAVE command to identify a specific set of saved monitor options. If no name is included, the saved set of options will be restored at the next system startup. The ABO, EXIT and SHUTDOWN commands include an automatic SAVEALL. The saved states of all options and file specs can be restored at any time by issuing the RESTORE command.

Table 2-2 (continued)

Command	Function
SAVER	A variation of the SAVE command which saves the currently displayed monitor page as a runnable batch file and appends a #RUN command to the end of the file. The same rules as those for the SAVE command apply for naming the file. Running a batch file created with SAVER, however, causes the monitor page to be filled-in and then the mode is run automatically.
SHU or SHUTDOWN	Exits from the GenRad monitor and returns system control to the RSX-11M operating system. (Same as ABORT, EXIT)
TIM or TIME	Displays the current date and time on the video screen.

UA, UB, and UC	User-assignable command names. They can be assigned to single or multiple monitor commands which can then be executed with the user-assigned command name. To assign commands, type the user-assigned name (UA, UB, or UC), followed by a space, and then the monitor commands. For example, to assign the name UA to the monitor's TIME command, type
	`#UA TIME`
	To execute the command, enter
	`#UA`
	and the date and time are displayed on the screen. To show what commands each name is assigned to, enter the user-assigned name followed by an equal sign. Using the previous example, typing
	`#UA=`
	results in the system responding with
	`#TIME`
	User-assigned names can contain complex monitor commands. The name can be assigned to a command string after the command has been entered to the monitor and tried. For example, if the user enters
	`#DIR SC1:*.OBC;*,.IDD;*`
	the system lists on the video display all files on device SC1: which have extension names .OBC or .IDD. To assign a name to the preceding command string, press CTRL R to restore the command line to the screen. The command line appears exactly as typed on the video display:
	`#DIR SC1:*.OBC;*,.IDD;*`

Table 2-2 (continued)

Command — **Function**

UA, UB, and UC (cont)

The cursor is automatically placed under the first letter of the command string, so typing the user-assigned name inserts the name before the command, shifting the command to the right. Do not forget to type a space between the user-assigned name and the command. Inserting the name UB and a space causes the line to look like this:

```
#UB DIR SC1:*.OBC;*,.IDD;*
```

Pressing the RETURN key completes the assignment of the user-name to the command string. The user can now list the SC1: device's .OBC and .IDD files by entering only

```
#UB
```

A user-name can be assigned to multiple monitor commands separated by a sharp sign (#); they must be on one line. For example, entering

```
#UC DIA#TEST#LOG=TIMES#RUN
```

assigns the commands to allow the user to run the Test sub-mode of Diagnose mode with the "Logging" option set to TIMES, by entering only

```
#UC
```

These user-assigned command names can be reassigned to different commands at any time in the same way they were assigned initially, that is, typing

```
#UA EDIT BIGBOARD.TPG/FILE=EXISTING
```

redefines the UA command to call the Editor instead of TIME. User-assigned commands are initially null at startup and can be reset to the null state by entering the user-name followed by only a space, without any commands. In the following example, the non-printing space character is represented by an underline, to indicate the presence of the space character. Entering

```
#UA_
```

assigns the name UA to a null string. Now, typing UA causes no response from the system.

UNLIMIT

Enables the monitor's UNLIMITed access level, allowing all monitor modes and utilities to be called and run. Entering

```
#UNLIMIT
```

causes the system to respond with the prompt

```
#Password?
```

Entering the proper password enables the UNLIMITed access level; the request is ignored if the password is not correct.

Table 2-2 (continued)

Command	Function
VTTRM	Sets the terminal display characteristics to those required by a VT10X terminal (normally set to correct value). Underlines or reverse video is used to indicate selected options, and screen editing can be used from VTTRM. (Related commands: GRTRM, HCTRM.)

Change-Page Commands

Change-page commands are used to call another major mode or utility options page from any other options page. Execute a change page command by entering the name of the major mode or utility page on the command line and pressing RETURN. The full name of the page is not required, only enough letters are needed to avoid ambiguity between names.

For example, to call the Editor mode, type

```
#E
```

Only the first letter is needed because no other modes begin with E.

Command-Line Directives

Task and utility options pages can be filled-in using either of two methods, screen-editing or command-line directives. Instructions for screen-editing are presented in paragraph 2.2.3.

Command-line directives can be used to change the setting or value of any option on the page. The form of the command to change options is:

Format

```
OPTION-KEYWORD = OPTION-VALUE
```

where:	OPTION-KEYWORD	Name of option field or file spec. Need only to use enough letters to distinguish between option names.
	=	Assignment operator
	OPTION-VALUE	Desired fill-in response or selection for the option.

Command-line directives are distinquished from change-page commands by the presence of the assignment operator (=) in the command-line directive entry.

An option value is checked for correct syntax before it is entered.

Multiple directives can be issued on one line by separating the directives by the symbol # (the prompt symbol).

When a MODE is being selected, the word MODE is not needed.

Examples

```
#MODE=TEST  (can be abbreviated to #TE)

#FIXTURE-ID=REPLACE  (can be abbreviated to #FI=R)

#MESSAGES=FILE#MES=SC1:SMPL1  (abbreviated #MESS=F#MES=SC1:SMPL1)
```

Command line directives are best suited as commands in batch files and in making small changes to the setup of a monitor page.

Single-Line Commands

A single-line command syntax allows tasks and utilities to be run from the command mode on one options page, with a minimum of typing and without going to another page. This is especially useful for running utilities, which have few enough options that it is inconvenient to have to fill in a page before running them. The current options page will be displayed again after executing a single-line command, and the monitor will still be in the mode from which the single-line command was entered. Only an explicit change-page command will change the current major mode or utility page.

Single-line commands are identified by including arguments (additional characters) after the page name on the command line. Typing only a page name without arguments is a change-page command when executed with RETURN, and the options page for the named major mode or utility is displayed. A page name without arguments can, however, be a single-line command if it is executed by pressing RUN instead of RETURN.

Options on the page of the mode to be run can be filled-in by including OPTION-KEYWORD=OPTION-VALUE arguments in the single-line command. Multiple arguments are specified at the end of the command line, separated by a slash character. Arguments without a slash character are considered to be file specifications. Input files are separated from output files on the command line by a greater-than sign (>), a less-than sign (<), or an equal sign (=). The direction of assignment for each symbol is:

Input > Output	(using greater-than sign)
Output < Input	(using less-than sign)
Output = Input	(using equal sign)

Examples of Single-Line Commands

```
#DIR SC1:1731.*;* > TI:/LIS=BR
```

Comments: Valid single-line command. File specification SC1:1731.*;* is entered for the Input file, and the Output file is set to TI: (the video display). The "Listing format" option is set to BRIEF (LIS=BR).

```
#DIR
```

Comments: Valid single-line command when entered with RUN. There are no arguments on the command line, but the command was entered with RUN instead of RETURN. This command will run the Directory utility exactly as options are set on the Directory page and then return the current page to the screen. Using RUN to enter the command line is especially useful for repeating a command.

2.2.3 Screen Editing

Screen editing is often the easier way to select options on the page. Entering any of the screen-editing commands from the command line causes an exit from command mode (the # prompt will be erased). The cursor will move to the selected option and changes can be made. Commands for moving from one option to another and for editing the option values are similar to standard GenRad editor commands (used in the TECO-macro and ED2 editors).

The cursor can be moved only to valid option fields on the screen. When moved to a fill-in type option, the cursor is at the left of the option field. When moved to a multiple choice type of option, the cursor is under the enabled setting, either the default or the option value previously set by the user.

Cursor Movement Path

When screen-editing is enabled and the cursor is on one of the options on the page, the cursor is moved ahead to the next option by pressing RETURN, the down-arrow key, CTRL N or TAB. To move the cursor back to the previous option press the up-arrow key or CTRL H. If an option-value is not syntactically correct, the cursor will not move from the option. The option field locations on a page are shown in Figure 2-2; the movement path in order from top to bottom is:

1. The top of the page ("NAME =" option)
2. The "DEV =" option
3. The left column of options
4. The right column of options (if any)
5. The left column of filespec options
6. The right column of filespec options (if any)

After this path has been completed, command mode is enabled and the cursor is placed on the command line. CTRL H or the up-arrow key moves the cursor along the same path in the reverse order.

Options Page Description

PAGE-NAME
PAGE-NAME, in the upper left corner of the options page, indicates which major mode or utility options page is present. The name in this field will change whenever a different major mode or utility is called.

NAME
The NAME supplied in this option is assumed to be the name for input files and is used as the name for output files created by the major mode or utility. The name typically describes the unit-under-test (UUT). The NAME is a system-wide default, appearing on all options pages; changing the NAME on one page will change it for all pages. The name used for specific files can be changed for each individual file by supplying the alternate name in the file specification options near the bottom of the page.

DEVICE
The DEVICE specified in this option is the disk device on which input files are located and where output files will be written. The DEVICE is a system-wide default, appearing on all options pages; changing it on one page will change it on all pages. The disk device used for specific files can be

changed for each individual file by including the alternate device in the file specification options.

Target System
Indicates the system for which software is being generated or the system that is running the test programs. The target system selection is made on the SETUP options page. Once a selection is made, it remains selected until explicitly changed. The indication appears on all monitor pages.

Options
Monitor page options are given two columns for display. The second column is used only when there are too many options to fit in one column. When a MODE option appears on a monitor page, it is the top-most option in the left column. Only the applicable options appear for each selected MODE. The MODE can be selected by typing only the MODE name, such as #TE rather than #MODE=TEST.

File specifications
File specifications are also listed in two columns; the input files are on the left, the output files on the right. Devices, UICs, names, extensions and versions can be explicitly assigned to most input and output files to override the system defaults. For "Temp" or "Intermediate" file spec option, the names and extensions are determined by the system. Only devices and UICs can be specified for these two types of files.

Command line/messages
The command mode is active when the sharp-sign prompt (#) is present near the bottom of the page. Three lines are reserved for help messages on specific options, error messages and the command line.

Screen-Editing Commands

Screen-editing is performed using commands to move the cursor to different options, and different selections within an option, and to edit fill-in fields. Other screen-editing commands save and restore option values, search for characters or keywords, display help text, and return control to command mode. Some screen-editing commands can be entered from the command prompt (#). Some commands cause the indicated edit action to take place in either the option field or in the command line, depending on where the cursor is at the time. Actions and the commands that perform them are listed in Table 2-3.

The simplest method of filling-in an options page is to move to the top with CTRL T, to move down the columns of options by pressing RETURN, and to step through option selections with the spacebar. Pressing RETURN, from the last option, enables the command mode, and the cursor is placed on the command line.

Selecting Multiple Choice Options

Enabling a selection of a multiple-choice option is accomplished by moving the cursor to the desired selection and then leaving the option. On the GenRad video display, if the default (leftmost) choice is selected, it is underlined; if the default is not selected, the selected option is shown in reverse video. On a VT10X, any selected option is underlined or shown in reverse video, depending on the terminal characteristic that is set. Hardcopy terminals display the enabled selection in capital letters and all other selections in small letters. Leaving the option is done by moving to another option or returning to the command mode.

Table 2-3
SCREEN-EDITING COMMANDS

Any Keyboard	GenRad Keyboard	VT10X Numeric Keypad	Cursor Movement or Editing Action
CTRL A		3	Move to left-most choice or character in option or command line.
CTRL B, ←			Move left one choice or character in option or command line.
CTRL C	MARK	PF2	Return to command prompt (#).
CTRL D		5	Delete one character to right in option or command line.
CTRL E			Move to right-most choice or character in option or command line.
CTRL F, →, spacebar			Move right one choice or character in option.
CTRL G			Same as CTRL C.
CTRL H, ↑		4	Move to previous option or, from #, to the last option.
CTRL I, ↓, TAB		0	Move to next option or, from #, to the first option.
CTRL J	RUN	LINE FEED	Run task from # or from an option.
CTRL K			Delete from cursor to end of option or command line.
CTRL L			Rewrite screen.
CTRL M, RETURN			Move to next option; does not function at #.
CTRL N			Same as CTRL I.
CTRL P			Character search in option or command line.
CTRL R			Restore previous value in option or command line.
CTRL T			Move to first option from any option or from #.
CTRL U			Delete line from start to cursor in option or command line.
CTRL V			Search and replace characters.
CTRL Z			Move to last option from any option or from #.
	SHIFT SRCH	ENTER	Keyword search from any option or from #.
	SRCH	. (dot)	Execute search or repeat search.
	DEL CHAR	DELETE	Delete one character to left in option or command line.
	SAVE	PF1	Save current option value.
	RESTORE	PF3	Restore current option value.
	HELP	PF4	Display monitor help text.
	SHIFT HELP	-	Display system help text.

When the spacebar is used to toggle multiple choice options, the cursor moves one selection to the right each time the spacebar is pressed. When the right end is reached, the cursor will wrap-around to the left-most selection. Note that using CTRL F or the right-arrow key and CTRL B or the left-arrow key does not wrap the cursor around but only bumps the right and left boundaries of the field and stays there.

Repeat Command Form

A numeric argument can be included to repeat the movement commands. Press the ESC key, followed by the number of times (n) to repeat the movement, and then the command. The repeat form of a command is

```
ESC n command
```

Searches

Any option field on the page can be accessed directly by searching for a keyword that matches the option name. A prompt for the keyword appears at the bottom of the screen when one of the search commands (SHIFT SRCH or ENTER) is entered. The search for a matching option field begins from the current cursor position and searches down the page, wrapping around to the top from the last file specification.

The character-search command, CTRL P, searches within an option for a selection beginning with the matching character. When a character-search command is entered, a prompt for the character appears at the bottom of the page.

A search-and-replace command, CTRL V, searches for a matching character or string in a fill-in type option and replaces the matching character(s) with character(s) supplied by the user.

Searches initiated from the command mode (refer to Command Line Directives, described earlier in this paragraph) will begin at the top ("NAME =") option. If a search initiated from the screen-editing mode fails, the cursor does not move from the current option. If a search initiated from command mode fails, the cursor remains on the command line.

2.2.4 Restore Feature

The Restore feature is useful if trouble is encountered when filling in an option. In screen-editing mode, when the cursor is moved to an option line (with SRCH, RETURN, CTRL N, etc), the existing entry is remembered by the monitor. If the user makes changes to a fill-in type option and is unable to make the entry acceptable to the monitor, pressing CTRL R will restore the field to what it was before the changes were made. Refer to para 2.2.5 for syntax checking. Changes can then be attempted again, or the restored value can remain in the option. Note that CTRL R is valid only while the cursor remains in the option field. Once the cursor is moved to a different option, CTRL R's memory is changed to be the existing value of that option.

When options are set using command-line directives, the Restore feature remembers the command line, and pressing CTRL R restores the entry on the command line exactly as it was entered. The command can then be edited and entered again. This is particularly helpful when an error occurs in a long command string. The line can be restored, edited, and resubmitted without the user having to retype the entire command string again.

The Restore feature of the monitor can be used when fill-in type options are being set to allow quick editing of errors in the command-line directive. After an option has been set, and before the next command is entered, the option directive can be restored to the command line by pressing CTRL R. The command, just as it was entered, reappears and can be edited. For example, consider the following entry for an input file specification:

```
#IN = SC1:1731501.TPH
```

The typographical error (TPH instead of TPG) can be corrected with 4 key-strokes:

```
CTRL R   CTRL E <DELETE> G
  ↑        ↑       ↑     ↑
  │        │       │     └──4) insert correct character (G)
  │        │       └──3) delete incorrect character (H)
  │        └──2) move to end of entry
  └──1) restore the command just as it was entered to the command line
```

2.2.5 Error Messages

Syntax checking of an option entry is done as soon as the entry is made. If there is an error in the entry, the system will not allow the user to move the cursor to another option. A message identifying the error will be displayed. When a character is then typed to correct the error, the message disappears.

The monitor will attempt to verify that input files needed by a mode (such as the .TPG file for the translator) are there; it will not call the function unless required input files are present.

If the syntax error occurred when the entry was being made from the command line, the error message replaces the entry on the display. The restore command will then return the command line to the display for editing.

2.2.6 HELP Facilities

Help facilities are available at all times for the user. Three forms of help are made available: monitor page help, specific help about an option, and General system help. There is no Help utility page; Help works on all major mode and utility pages.

Monitor Page Help

General help is requested from the command mode by typing HELP from the command line or pressing the HELP key on the keyboard or the PF4 key on the VT10X terminal. The first time this command is entered, information about the current monitor options page is displayed (without erasing the screen). Pressing the spacebar will step through a series of entire pages giving information first about the current monitor mode and then about the entire system. Pressing the RETURN key at any time returns control to the mode which was active before help was requested.

Specific Option Help

Specific help can be requested from both command mode and screen-editing mode. While screen editing is being used, moving the cursor to an option and pressing the HELP key will put information in the 3 lines at the bottom of the screen reserved for help-text, command lines and error messages. From the command mode, help about a specific option is requested by specifying the option name in the HELP command. The message will describe both the option and the specific current selection. For example, with the option:

```
Type of listing [ERRORS_ONLY,FULL,NONE]
```

Placing the cursor on the option and pressing the HELP key, or entering the command

```
#HELP TYPE
```

results in the following help message being displayed on the screen:

```
No listing (.LIS) file is produced when NONE is specified.
```

Note that the message would be different for selections of ERRORS_ONLY or FULL. Filespec help will include the full file to be created or used. For example, if the following options are specified

```
NAME = 1731, DEV = SC1:, TPG = .DTP
```

the help-text for the TPG = option reads:

```
Name of Test Program source file.
FILE SPEC =  SC1:1731.DTP
```

The help-text displayed will change when the cursor is moved from one option to another and when different selections on the option line are enabled. This Help mode will remain on, displaying messages as each new option is encountered, until it is turned off by pressing the HELP key again. Leaving the screen-editing mode will also turn off the Help mode.

System Help

General help on using the system is displayed by pressing the SHIFT HELP key on the GR keyboard or the _ key on the VT10X keypad.

2.2.7 Running the Mode

After the required options on the page have been filled-in, run the task or utility by pressing the RUN key or typing the RUN command. For all monitor modes except Diagnose, this will perform the task or utility function as specified on the options page. Entering RUN to the Diagnose mode page loads the test programs and enables the lower-level monitor for the Run-Time System (RTS). The RTS provides the software control of the system's test instruments (hardware) and is active while the system is being used for program debugging or board testing.

File Management – Section 3

3.1 INTRODUCTION

Test system software consists of many system and user files. All these files are stored on magnetic disks. Disk devices are file-structured devices, which means that files are accessed by names which are automatically indexed by the device controller. All other system physical devices are serial devices because data are input and output in the same order. These devices are the keyboard, keypad, video display, strip printer, and optional line printer.

Disk drive configurations, shown in Table 3-1, vary between models of the 227X series of workstations.

Table 3-1
227X DISK DRIVE CONFIGURATIONS

System	Drive	Devices	Storage	Purpose
2270	Hawk	2 Fixed	4.9 Mbytes	System Software
		2 Cartridge	4.9 Mbytes	User Programs & Backup
	Floppy	2 Disks		Loading Software
2271/72	Phoenix	3 Fixed	42 Mbytes	System Software
		1 Cartridge	14 Mbytes	User Programs & Backup
	Floppy	2 Disks		Loading Software
2275	Winchester	1 Fixed	21 Mbytes	System & User Software
	Floppy	1 Disk		Loading Software & Backup

A 2270 with the Foreground/Background option has a second Hawk drive.
A 2275 may have an optional second Winchester drive.

3.2 SPECIFYING DEVICES

3.2.1 Introduction

Physical disk surfaces, called physical devices, are not an especially practical way of referring to locations for storing files. The file management part of the system software has an automatic way of assigning various areas of the physical devices to files in accordance with the type of file. These areas are called logical devices. There may be several logical devices on each physical device and the size of logical devices varies with usage. File specifications that use the logical device names will be automatically indexed by file type.

Logical device names also allow users on multi-user stations to have their own standard devices, thus freeing them from having to know which physical device is assigned to a particular terminal. For example, assigning OUTPUT=TI:

from the foreground automatically selects the foreground console; assigning OUTPUT=TI: from the background automatically selects the VT10X terminal.

Logical device names make it easier for users to work on various systems that have different numbers and types of disk drives. For instance, files designated to be stored on logical device SC: can always be found, regardless on which system the file is installed; but, the physical device FW2: would not be recognized by any system except a 2275. Systems that have a removable disk cartridge use that cartridge device for the SC: logical device and the AL: logical device. Removable devices are also referred to as the port. The rationale for these names is as follows:

> SC: refers to scratch device, which implies user-generated files.
>
> AL: refers to allocatable device, which implies assignable to specific users and user-mountable device.
>
> Port implies access to external or removable device.

Therefore, the 3 terms may be used synonymously in many applications but do have slightly different meanings.

In single-user systems, SC: and AL: are actually synonymous. In multi-user systems, SC: refers to a single area of disk space that is shared by all users. AL: refers to a disk space allocated to just one user. On 2270 systems, where foreground and background users actually have different physical devices for AL:, the difference between SC: and AL: becomes significant. When using a disk utility (DCOpy, DCHange, or DFormat) on a 2270 multi-user system, it is recommended that file devices be AL:.

It is recommended that both SC: and SC1: be system scratch devices and not be used for permanent file storage. SR: and SR1: are available for medium-term storage of output files or test programs. It is also recommended that the foreground user store files on SR: and the background user store files on SR1:. For long-term storage of files, a separate removable disk pack should be used.

Tables 3-2, 3-3 and 3-4 show to which physical device the system assigns each logical device. The zero is not necessary in a device name. That is, SC: is used to mean SC0:.

3.2.2 2270 System With Hawk Drive

The Hawk disk drive and the associated dual floppy disk drive are configured for system use as shown in Figure 3-1. The Hawk drive is configured as 4 physical devices, DK0: through DK3:. All the devices are available for writing or reading, but they have certain conventional uses. DK0:/DK1: is the fixed master disk containing the operating system, libraries, and all other GR-supplied software. DK2:/DK3: is the removable disk cartridge normally used to store user-created files (circuit descriptions, test programs, etc). The floppy disk drives (DY0: and DY1:) are used to store test programs and to transfer and update software.

Multi-user (Foreground/Background) systems contain a second disk drive to provide adequate storage space for 2 users. The disk drive configuration for these systems is shown in Figure 3-2.

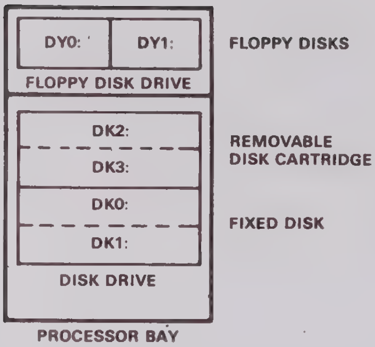

DY0:
DY1:
FLOPPY DISKS
FLOPPY DISK DRIVE
DK2:
DK3:
REMOVABLE DISK CARTRIDGE
DK0:
DK1:
FIXED DISK
DISK DRIVE
PROCESSOR BAY

Figure 3-1. 2270 Single-User-Station Disk Device Locations.

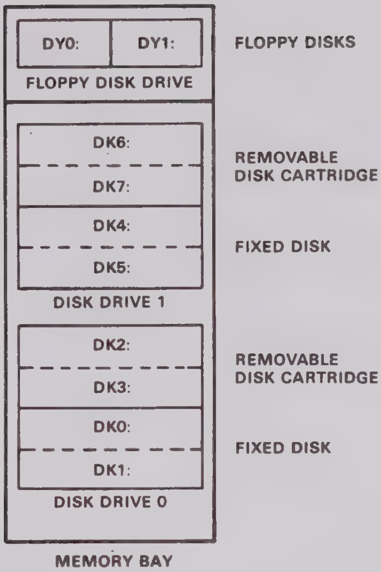

DY0:
DY1:
FLOPPY DISKS
FLOPPY DISK DRIVE
DK6:
DK7:
REMOVABLE DISK CARTRIDGE
DK4:
DK5:
FIXED DISK
DISK DRIVE 1
DK2:
DK3:
REMOVABLE DISK CARTRIDGE
DK0:
DK1:
FIXED DISK
DISK DRIVE 0
MEMORY BAY

Figure 3-2. 2270 Multi-User-Station Disk Device Locations.

Table 3-2
2270 PHYSICAL/LOGICAL DEVICE DESIGNATIONS

Logical Device	Physical Device	Recommended Usage
TI:	TT:	Standard (or foreground) GenRad terminal
TI:	TT1:	Background (VT10X) terminal (in multi-user systems)
LP:	LP:	Strip printer
LP1:	LP1:	Line printer (Spooler)
SY:	DK:	System and default device
MD:,MD1:	DK:,DK1:	Monitor system
LB:	DK:	RSX-11M operating system
LB1:	DK1:	Library files
LB2:	DK1:	GRnet interface for single-user system, future update libraries, storage for network files
LB2:	DK2:	GRnet interface for multi-user system
AL:	DK2:	Allocatable device (port) for single-user or foreground user files
AL:	DK6:	Allocatable device (port) for background user files on multi-user system

Table 3-2 (continued)

Logical Device	Physical Device	Recommended Usage
SC:	DK2:	Scratch and user program
AL1:	DK3:	Allocatable device (port) for single-user or foreground user files
AL1:	DK7:	Allocatable device (port) for background user files on multi-user system
SC1:	DK3:	Scratch and user programs
SR:	DK4:	User program disk for multi-user system
SR1:	DK5:	User program disk for multi-user system
--	DY:	Software support device
--	DY1:	Software support device
NL:	--	Null device
NT:	DK2:	Storage for networking tasks (229X)
ND:	DK2:	Network default device

3.2.3 2271/2272 Systems with Phoenix Drive

The Phoenix disk drive and the associated dual floppy disk drive are configured for system use as shown in Figure 3-3. The Phoenix drive is configured as four physical devices, DM: through DM3:. All the devices are available for writing or reading, but they have certain conventional uses. DM0: is the fixed master disk containing the operating system, libraries, and all other GR-supplied software. DM1: is the removable disk cartridge normally used to store user-created files (circuit descriptions, test programs, etc). DM2: and DM3: are fixed disks used for general-purpose storage. The floppy disk drives (DY: and DY1:) are used to store test programs and to transfer and update software.

The Phoenix disk drive has sufficient storage space to accommodate both single-user and multi-user systems (i.e., systems with the Foreground/Background option).

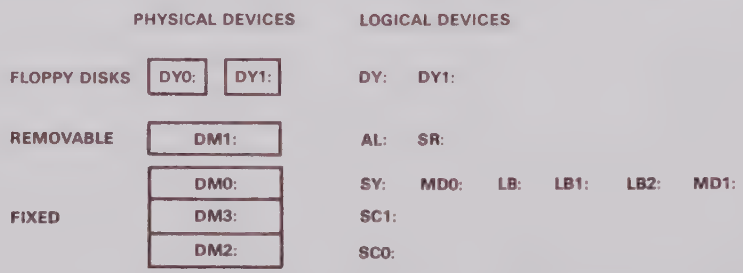


Figure 3-3. 2271/2272 Disk Drive Configuration.

Table 3-3
2271/2272 PHYSICAL/LOGICAL DEVICE DESIGNATIONS

Logical Device	Physical Device	Recommended Usage
TI:	TT:	Standard (or foreground) GenRad terminal
TI:	TT1:	Background (VT10X) terminal (in multi-user systems)
LP:	LP:	Strip printer
LP1:	LP1:	Line printer (Spooler)
SY:	DM:	System and default device
MD:,MD1:	DM:	Monitor system
LB:	DM:	RSX-11M operating system
LB1:	DM:	Library files
LB2:	DM:	GRnet interface, future update libraries, storage for network files
AL:,SR:	DM1:	Allocatable device (port) for user files
SC:	DM2:	Scratch and user programs
SC1:	DM3:	Scratch and user programs
--	DY:	Software support device
--	DY1:	Software support device
NL:	--	Null device
NT:	DM:	Storage for networking tasks (229X)
ND:	DM:	Network default device

3.2.4 2275 Workstations with Winchester Drive

The Winchester drive and the associated floppy disk drive are configured as shown in Figure 3-4. The Winchester drive and the floppy disk drive are each configured as a single physical device, FW2: and FW0:, respectively. All system software is permanently stored on FW2:. All user software is normally loaded on FW2: from an FW0: disk prior to operation. User software can be run directly from FW0:, but system default devices may not be effective. FW0: is used to transport user software, to install software updates, and to backup the software stored on FW2:.

A second Winchester is optionally available for storing more user software on-line and therefore reducing the need for loading user software. This device is assigned FW3:.

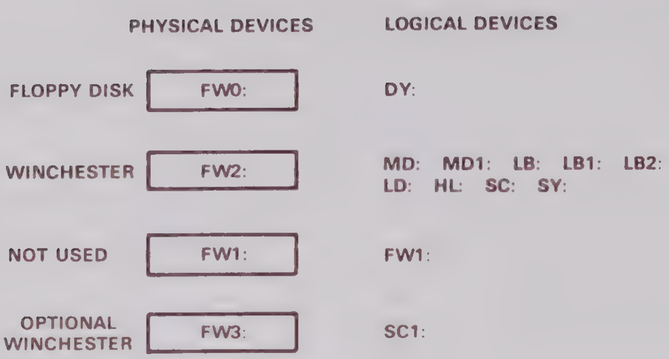

PHYSICAL DEVICES
LOGICAL DEVICES
FLOPPY DISK
FW0:
DY:
WINCHESTER
FW2:
MD: MD1: LB: LB1: LB2:
LD: HL: SC: SY:
NOT USED
FW1:
FW1:
OPTIONAL WINCHESTER
FW3:
SC1:

Figure 3-4. 2275 Device Drive Configurations.

Table 3-4
2275 PHYSICAL/LOGICAL DEVICE DESIGNATIONS

Logical Device	Physical Device	Recommended Usage
TI:	TT:	Standard (or foreground) terminal
TI:	TT1:	Background (VT10X) terminal (in multi-user system)
LP:	LP:	Strip printer
LP1:	LP1:	Line printer (Spooler)
DY:	FW0:	Software support device
SY:	FW2:	System and default device
MD:, MD1:	FW2:	Monitor system
LB:	FW2:	RSX-11M operating system
LB1:	FW2:	Library files
LB2:	FW2:	GRnet interface and Update libraries
LD:	FW2:	Load device
HL:	FW2:	Help files
SC:	FW2:	Scratch space and user programs
SC1:	FW3:	Optional scratch space and user programs
NL:	--	Null device
NT:	FW2:	Storage for networking tasks (229X)
ND:	FW2:	Network default device

3.2.5 Transporting Software between 227X Systems

The method of moving files from one system to another varies, depending on the models involved. With systems of the same model, the move is made by taking the removable disk from one and inserting it in the other. This is also true when moving between a 2271 and a 2272 since the movable mass storage media is the same. The 2270, 2271, and 2272 all have identical floppy disk drives, so when moving between a 2270 and a 2271 or a 2272, the files can be transferred to floppy disks and moved to the other system. The floppy disk drive on a 2275 is different from those on a 2270/71/72. The normal recording format on the 2275 is double-sided. The recording format on a 2270/71/72 is single-sided, RX02. The format is selectable on a 2275 so that a 2270/71/72 format can be used to transfer files between a 2275 and a 2270/71/72.

One other alternative is to make use of the optional network facilities for direct transfer of files between any 227X and/or 229X system.

3.3 SPECIFYING FILES

Files are specified in terms of 5 identifiers in the following format:

```
DEV:[UIC]NAME.EXT;VERSION
```

where:	DEV	Device name as shown in Tables 3-2, 3-3, or 3-4. If DEV: is not specified, the default designation SY: is assumed. Logical device names should be used for all file specifications.
	[UIC]	The RSX-11M Operating System manages tasks and file space by User Identification Code (UIC). When a user logs-on to the system, the UIC must be provided. Generally the user of a 227X Workstation does not need to know the UIC because the automatic initialization features takes care of supplying it to RSX-11M. For reference, the UIC used is [200,200]. There are a few occasions when a user may need to know more about using UICs. Refer to the 227X System Manager's Guide.
	NAME	File name consisting of up to 9 alphanumeric characters.
	EXT	File extension consisting of up to 3 alphanumeric characters. Some monitor modes automatically assign the file extension if none is specified. Table 3-5 lists these extensions.
	VERSION	An octal number up to 32767. The system automatically updates this number each time a new version of a file is created. When the version number is omitted, the system assumes that the highest version number file has been selected. This default to the highest number can be overridden by explicitly specifying the version number desired. For example, SY:PCBRD.TPG;6 causes the system to access version 6 of the file. Specifying version number 0 (e.g., SC:PCBRD.TPG;0) causes the system to access the highest version of the specified file. Version numbers are a convenient way of maintaining multiple versions of a file on a disk. However, version numbers should not be relied upon to track the most current file because they could get out of sequence when moving files between devices.

In some cases there may be a need to refer to all files in a particular category. For example, all versions of the file, BOARD1, located on SC: with extension .TPG may need to be listed. Instead of specifying each file separately, a "wild-card" character (*) is provided which may be substituted for the version specification to indicate all versions, as shown below:

```
SC:BOARD1.TPG;*
```

Typically, the wild-card character (*) is used in file management modes for any of the 4 file attributes (UIC, name, extension, version) when applicable.

Table 3-5
DEFAULT EXTENSIONS FOR FILES CREATED AND UPDATED BY SYSTEM TASKS

Task	Created File Type	Extension
CKTGEN	Circuit Description File	.CKT
	CKTGEN data file	.CIR
	CKTGEN error listing	.ERC
	CKTGEN report file	.CRP
	Nail data base file	.NDB
ATG	Analog Test Program	.ATP
	Continuity Test Program	.CTP
	Digital Test Program	.DTP
	Fixture Test Program	.FTP
	Merged Test Program Source	.TPG
	.TPG file with temporary nails	.TPX
	Report of Analog ATG Results	.RPT
	Circuit Description Extractor Report	.ERR
	Cross Reference Report	.AXR
	ATG Internal Work File	.WOR
	Digital ATG Error Listing	.ERP
	Digital ATG Report of Results	.MSG
	In-Circuit Diagnostic Data File	.IDD
	.IDD file with temporary nails	.IDX
	Digital ATG Interconnect File	.DAI
Translate	Object Code	.OBC
	Listing	.LIS
	Symbol Table	.SMT
Library Management	Analog Library Object	.LIB
	Analog Library Source	.LBS
	Library Listing	.LBL
	Digital Library Source	.DTS
	Digital Library Object	.DTL
	Component Library Source	.ACS
	Component Library Object	.ACL
Batch	Input File	.BCH
	Output File	.BLI
Diagnose	Debug Trace File	.DBT
	Logging File	.LOG
	Message File	.MES

Table 3-5 (Continued)

Task	Created File Type	Extension
Modify	Incremental Change File	.INC
	Incremental Listing File	.ITL
Nail Assignment	Nail Assignment Report	.NAR
	Nail Data Base	.NDB
	Nail Fixture Report	.NFR
	Nail Assignment Status	.NLS
	Test Program Source	.TPG
	In-Circuit Diagnostic Data	.IDD
	Automatic Test Options	.ATO
	Circuit Description File	.CKT
User/Editor	Automatic Test Options	.ATO
	.ATO file with temporary nails	.ATX
UPDATE	.TPG with Debug changes	.TPG
	Updated .TPG when errors exist	.TPU
	Final version of .DBT	.DBO
REPORT GENERATOR	Logging Current Statistics File	.LOG
	Summary Statistics File	.SUM
	Report File	.REP

3.4 MANAGING FILES AND DEVICES

The managing of files consists of procedures that are accomplished through the use of the various utilities that exist in the monitor. Paragraph 3.5 provides, in alphabetical order, a detailed description of each utility and examples of how they are used. Many procedures require the use of several utilities. Paragraphs 3.4.1 and 3.4.2 present examples of procedures that use a series of utilities. Each example states the objective of the procedure and then lists the utilities required, in the appropriate order.

For details on creating and editing files, refer to Section 11.

For details on managing library files, refer to the 227X Test Library Programming manual.

3.4.1 Combining Files

Assume there exists an .ATO file for PCBA and another for PCBB and you want to combine them into a new .ATO file for PCBB. All existing and new files should be on device SC:. The utilities required are:

DIRECTORY	Check to see that files are on SC:.
COPY	Combine 2 existing files.
EDITOR	Remove unwanted END statement.
PURGE	Delete old versions of .ATO files.
PRINT	Print the file on printer.

3.4.2 Copy Files on a Floppy Disk to a Cartridge

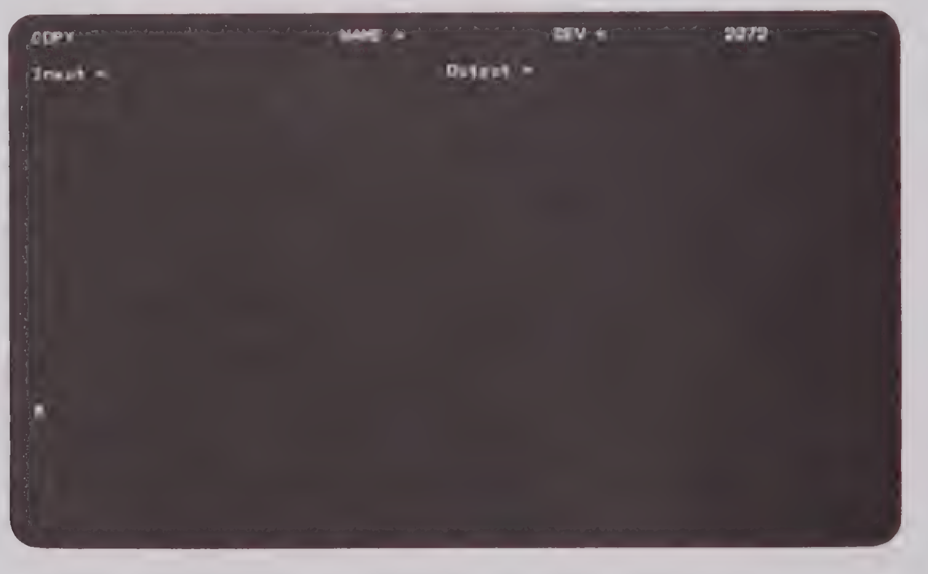
You have a floppy disk that contains only a set of files pertaining to a particular pc board. You want to copy this whole floppy, if it will fit, to the removable cartridge disk that is currently mounted on the test system. The utilities required are:

FREE	Check to see if adequate space is available on cartridge device.
DCHANGE	Necessary if inadequate space and new cartridge is to be installed.
DFORMAT	Necessary if a new unformatted cartridge is installed.
DCHANGE	Install the floppy disk.
COPY	Copy files from floppy to cartridge.

3.5 MONITOR UTILITY DESCRIPTIONS

3.5.1 Copy Utility

The Copy utility provides an easy method for duplicating files, moving files to a different device, or merging several files into one larger file. Before copying a file, ensure there is enough room on the device to which the file will be copied. Insufficient space could pose a problem especially when a large file is being copied from a hard-disk to a floppy. Use the Free utility to determine the number of free blocks on the destination device. The Copy options page, shown in Figure 3-5, is displayed when CO is typed.



```
COPY                    NAME =          DEV =           2272
Input =                         Output =
```

Figure 3-5. Copy Utility Options Page.

Input =

The minimum required entry for this file spec is the extension name of the file to be copied. If the file is not on the device in the "DEV =" option, then the entry must include the proper disk device. If the name of the file to be copied is different from the "NAME =" entry, the name must be included in this entry also. Multiple file specs, separated by commas, can be entered as input files to be combined into a single output file.

Output =

The name and extension for the file to be created do not default to anything and must be specified here. The "DEV =" option is assumed if no alternate device is included.

Examples

To copy the .CKT file for board PCBI from SC: to AL:, type

```
#CO
```

to display the option page. Specify the options as follows:

```
NAME = PCBI
DEV = SC:
Input = .CKT
Output = AL:PCBI.CKT
```

To execute the mode, press RUN key or type RUN. If there is already a PCBI.CKT file on the AL: device, the copied file will be assigned the next higher version number; otherwise it will be assigned version 1.

The command line method of copying the files in the above example is to type the following command after the prompt (#) on any page:

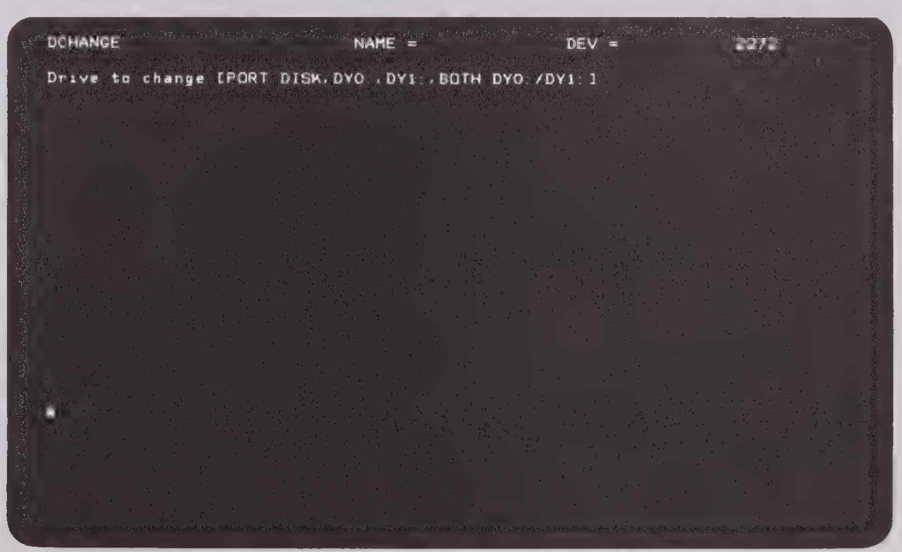
```
#COPY .CKT > AL:PCB1.CKT
```

To copy several files into one file, the command would have the form

```
#COPY A.BCH,B.BCH,C.BCH > FULL.BCH
```

3.5.2 Dchange Utility

The Dchange utility provides a means of changing the disk media in the port disk device or the floppy drives without having to reboot the system. An options pages is displayed by typing DCH. The 2275 disk devices are different from those on a 2270/71/72. The software automatically displays the appropriate page for the target system specified. See Figures 3-6a and 3-6b.



```
DCHANGE                    NAME =             DEV =          2272
Drive to change [PORT DISK,DY0:,DY1:,BOTH DY0:/DY1:]
```

Figure 3-6a. Dchange Utility Options Page for the 2270/71/72 Workstations.

Drive to change [PORT DISK,DY0:,DY1:,BOTH DY0:/DY1:]

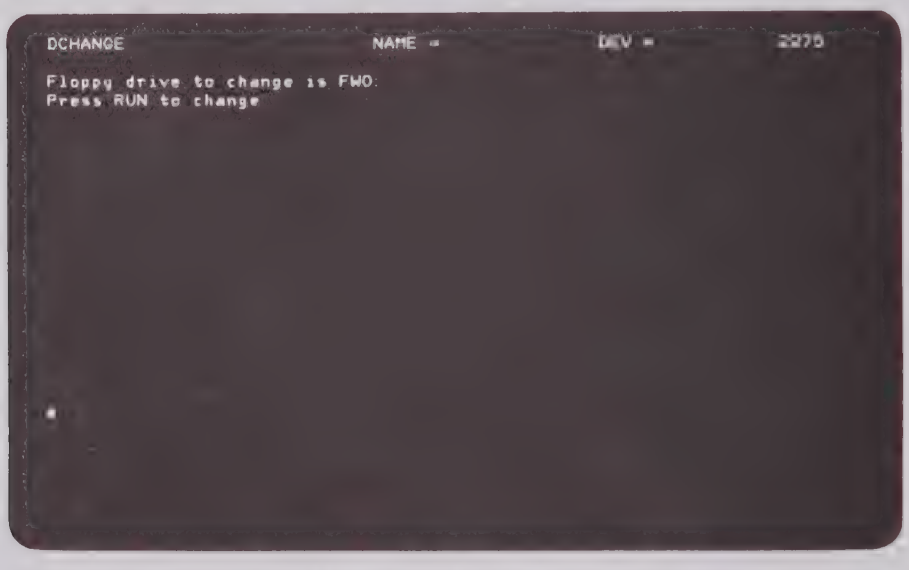
The drive(s) specified by this option will be software dismounted and can then be removed from the drive. The disk to be mounted is inserted into the drive and, when the RUN or START/CONTINUE button is pressed, the disk will be software mounted.

[PORT DISK] Selects the removable hard-disk cartridge as the disk to change. In 2270 multi-user systems, this is DK6/DK7 only.

[DY0:] Selects the left-side floppy disk in the DSD-440 disk-drive as the disk to change.

[DY1:] Selects the right-side floppy disk in the DSD-440 disk-drive as the disk to change.

[BOTH DY0:/DY1:] Selects both floppy disks in the DSD-440 disk-drive are to be changed.



```
DCHANGE                    NAME =            DEV =          2275

Floppy drive to change is FWO:
Press RUN to change
```

Figure 3-6b. Dchange Utility Options Page for the 2275 Workstation.

There are no options on the 2275 Dchange page because there is only the one floppy that can be changed. The note "Floppy drive to change is FW0:" is a reminder of the device name of the floppy.

Examples

To change the removable hard-disk cartridge on a 2270/71/72, type

```
#DCH /DRIVE=PORT
```

A message indicating that the disk has been dismounted appears, followed by the instructions

```
SPIN DOWN DISK AND SPIN UP PORT DRIVE WITH NEW DISK IN PLACE.
PRESS START/CONTINUE OR RUN ONLY WHEN DISK READY LIGHT IS ON.

PAUSING
```

Follow the cartridge removal and inserting procedures in the appropriate operation manual. Press START/CONTINUE or RUN after the READY light comes on. When loading is complete, the original monitor page will return to the display. In 2270 multi-user systems, spin down top drive only.

To change a floppy disk on a 2270/71/72, type

```
#DCH /DRIVE=DY0:
```

The following question is displayed:

```
Is DY0: going to have a new diskette loaded [Y/N]?
```

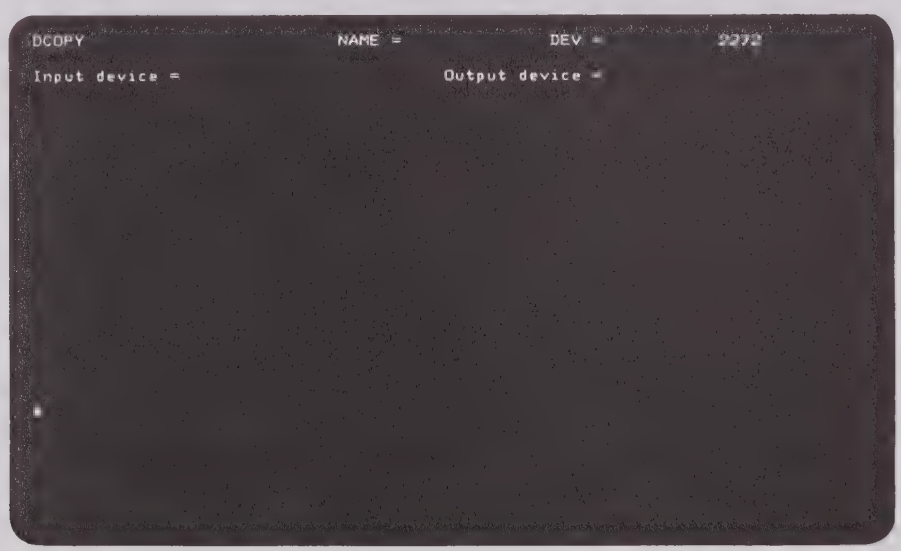
Typing N ends the Dchange run. Typing Y readies the system for a new diskette and displays the instructions

```
CHANGE FLOPPY DISKETTE(S) WHILE THE SYSTEM IS PAUSED THEN
PRESS START/CONTINUE OR RUN ONLY WHEN READY TO PROCEED.

PAUSING
```

Remove the old diskette, insert the new one, and press RUN to complete the operation.

3.5.3 Dcopy Utility

Dcopy provides a quick and easy method of duplicating the contents of one hard-disk or floppy onto another disk of the same type (e.g., hard-disk to hard-disk, floppy to floppy; NOT hard-disk to floppy or floppy to hard-disk). The options page, shown in Figure 3-7, is displayed by typing DCO.



```
DCOPY                    NAME =              DEV =          2272
Input device =                      Output device =
```

Figure 3-7. Dcopy Utility Options Page.

Input device =
This is the device containing the disk or diskette to be duplicated. As the entire disk surface is to be copied, enter only the device name. If no entry is made, the device will default to the DEV = option. Do not include file names or extensions in this option.

Output device =

This option specifies the device containing the disk or diskette on which the input device disk is to be duplicated. The output device must be of the same type as the input device. Copying the entire contents of a disk using Dcopy is not allowed from a hard disk to a floppy diskette or from a diskette to a disk. If no device is specified, it will default to the DEV = option. However, the same device cannot be specified for both input and output, so if the default is assumed for one device, another must be specified in the other file spec option.

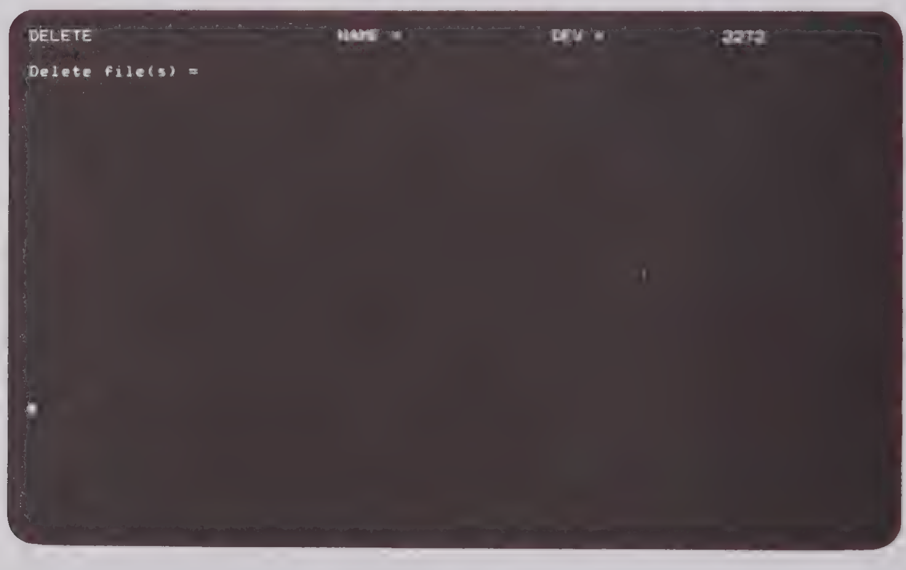
Examples

```
    #DCO /IN=SY:/OUT=SC:
or
    #DCO SC:=SY:
```

Either command copies all data on the fixed disk SY: to the removable disk SC:.

3.5.4 Delete Utility

The Delete utility provides a means of removing obsolete files from a disk device. Use the Delete utility with caution to avoid losing important files. The options page, shown in Figure 3-8, is displayed by typing DE.



```
DELETE                    NAME =          DEV =          2272
Delete file(s) =
```

Figure 3-8. Delete Utility Options Page.

Delete file(s) =

The minimum entry for this file specification is the extension and version number of the file to delete. The version number is required as a safety feature to reduce the chance of unintentionally deleting important files. The disk-device and file name will default to the DEV = and NAME = options, unless specifically included in this file spec. More than one file at a time may be deleted by entering a string of file specifications separated by commas. The use of wild cards, as described in para 3.3, is allowed; however, use extreme caution with wild cards for file spec fields (*.*;*) as this will delete ALL user files from the specified device.

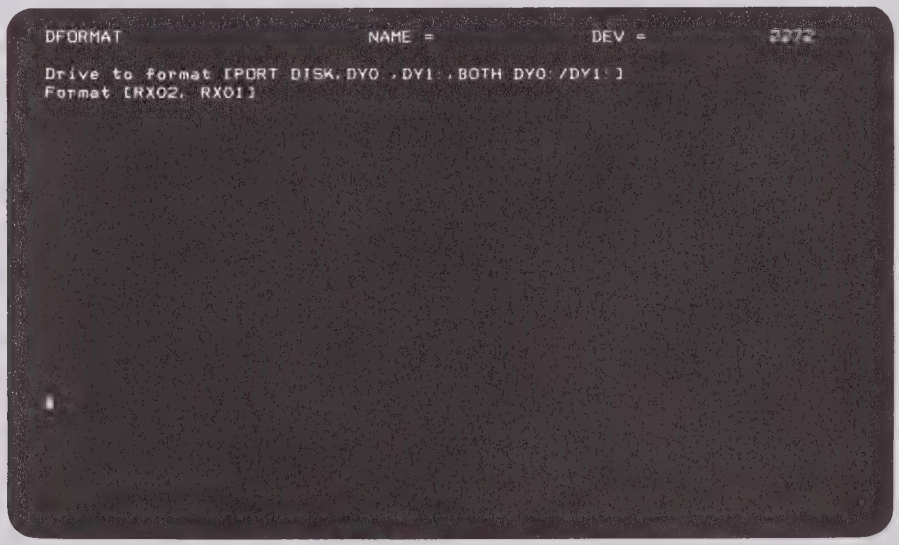
Example

```
#DEL SCRATCH.TPG;*
```

All versions of the file SCRATCH.TPG on the DEV= device are deleted.

3.5.5 Dformat Utility

A hard-disk cartridge or floppy diskette must be formatted and initialized before it can be used in the test system. The Dformat utility automatically formats and initializes the specified disk or diskette. Insert the disk or diskette to be formatted into a disk-drive, specify that drive in the "Drive to format" option, and run the utility. The change-page command DFORMAT (abbreviated DF) calls the utility, and an options page appears. The 2275 disk devices are different from those on a 2270/71/72. The software automatically displays the appropriate page for the target system specified. See Figures 3-9a and 3-9b.



```
DFORMAT                        NAME =              DEV =        2272

Drive to format [PORT DISK,DY0:,DY1:,BOTH DY0:/DY1:]
Format [RX02, RX01]
```

Figure 3-9a. Dformat Utility Options Page for 2270/71/72 Workstations.

Drive to format [PORT DISK,DY0:,DY1:,BOTH DY0:/DY1:]

[PORT DISK] Selects the removable hard-disk cartridge in the port drive as the disk to be formatted.

[DY0:] Selects the floppy diskette in the left-side of the DSD-440 disk drive as the diskette to format.

[DY1:] Selects the floppy diskette in the right-side of the DSD-440 disk drive as the diskette to format.

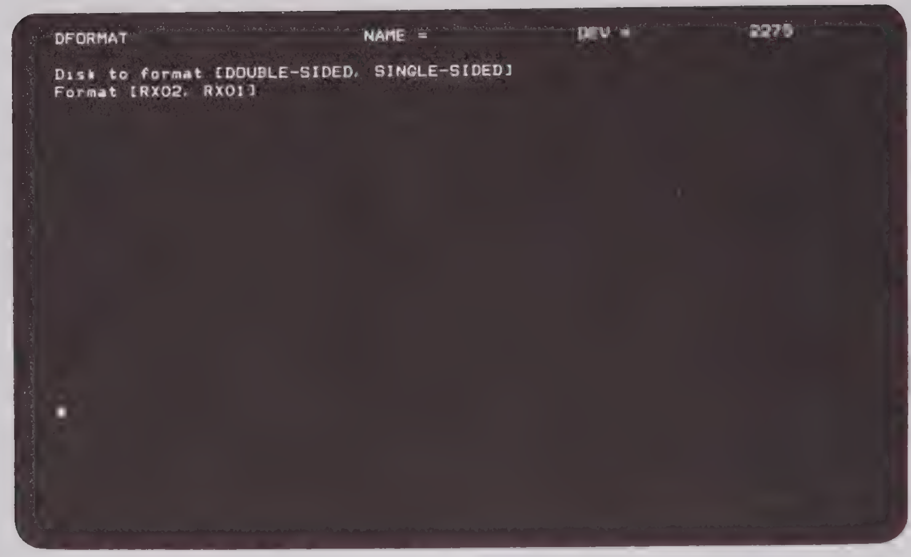
[BOTH DY0:/DY1:] Selects both floppy diskettes in the DSD-440 disk drive are to be formatted.

Format [RX02, RX01]

Appears when any of the DY options is selected.

[RX02] Selects double density format compatible with 2270/71/72.

[RX01] Selects single density format compatible with a DEC VAX.



```
DFORMAT                     NAME =               DEV =          2275

Disk to format [DOUBLE-SIDED, SINGLE-SIDED]
Format [RX02, RX01]
```

Figure 3-9b. Dformat Utility Options Page for 2275 Workstations.

Disk to format [DOUBLE-SIDED, SINGLE-SIDED]

[DOUBLE-SIDED]	Selects the formatting of the 2275 floppy for double-sided use. This is the normal operating usage.
[SINGLE-SIDED]	Selects the formatting of the floppy for single-sided use. This is done when floppies are to be transported between 2275 Systems and 2270/71/72 Systems.

Format [RX02, RX01]

Appears when the SINGLE-SIDED option is selected.

[RX02] Selects double density format compatible with 2270/71/72.

[RX01] Selects single density format compatible with a DEC VAX.

Examples

2270/71/72

To format a removable cartridge, select PORT DISK and press RUN. The system prompts for confirmation of intent by displaying

```
WARNING:  ALL DATA ON PORT DISK WILL BE LOST
              AS THE DISK IS FORMATTED

                 PROCEED [Y/N]:?
```

Typing N cancels the formatting operation and data on the port disk remains undisturbed. Typing Y erases, formats, and initializes the port disk cartridge.

To format the left floppy disk, select DY0: and press RUN. The system prompts for status of the other floppy disk:

```
Are you using the OTHER drive (is DY1: allocated and mounted) [Y/N]:?
```

Answer Y or N. The system then prompts for confirmation of intent by displaying

```
WARNING:  ALL DATA ON DY0: WILL BE LOST
          DURING FLOPPY DISKETTE FORMATTING.

               --PROCEED--[Y/N]:?
```

Typing N aborts the formatting. Typing Y displays the message

```
FLOPPY FORMATTING AND INITIALIZING IN PROGRESS -- DY0:
```

and completes the formatting task.

2275

To format a floppy disk for use on a 2275: select DOUBLE-SIDED, press RUN.

3.5.6 Directory Utility

The Directory utility produces a directory listing of the specified files on the specified disk-device, either hard-disk or floppy. Usually, the Directory utility is run from a single-line command submitted from any major mode page. Examples of a single-line command are:

```
#DIR AL:TRAINGBD.ERR;*>TI:/LIS=BR
```

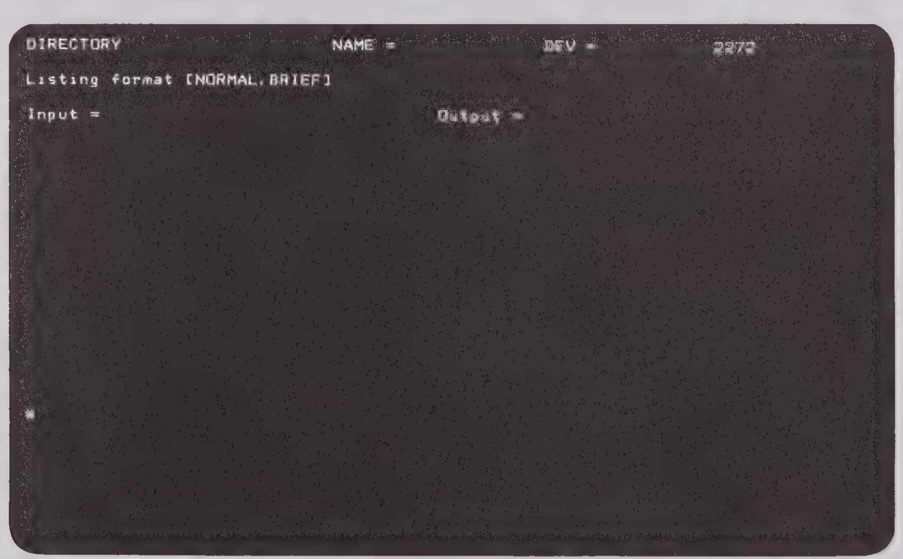
or

```
#DIR TI:=AL:TRAINGBD.ERR;*/LIS=BR
```

These are equivalent commands which do not rely on any defaults. They list, to the video display, a brief listing of all versions of the .ERR file for board name TRAINGBD that exist on device AL:.

The end of the directory listing contains a summary of files and disk space in blocks. The disk space includes 2 numbers. The first number is the number of blocks actually used and the second number is the number of blocks allocated for all the files in the list.

Typing DIRECTORY (DIR) to the command prompt with no file name or output device calls the Directory mode, and the page shown in Figure 3-10 appears.



```
DIRECTORY                      NAME =             DEV =            2272
Listing format [NORMAL,BRIEF]
Input =                                  Output =
```

Figure 3-10. Directory Utility Options Page.

Listing format [NORMAL,BRIEF]

[NORMAL] The default selection, NORMAL, produces a complete listing for the device, including file-names, extensions and version numbers, sizes of the files in blocks, and the dates of creation of each file.

[BRIEF] Enabling BRIEF produces a listing including only the file-names, extensions and version numbers of each file on the device.

Input =

The minimum entry required for this option is the extension(s) of the file(s) to list. The device and name default to "NAME =" and "DEV =". The wild card character (*) can be any of the file specification fields ([UIC]name.extension;version.)

Output =

Typically, "Output = TI:" is used to list the directory to the video display. Entering a file specification will write the directory to that file. Specifying "Output = LP0:" (or LP1: for system with the optional line printer) sends the directory to the printer.

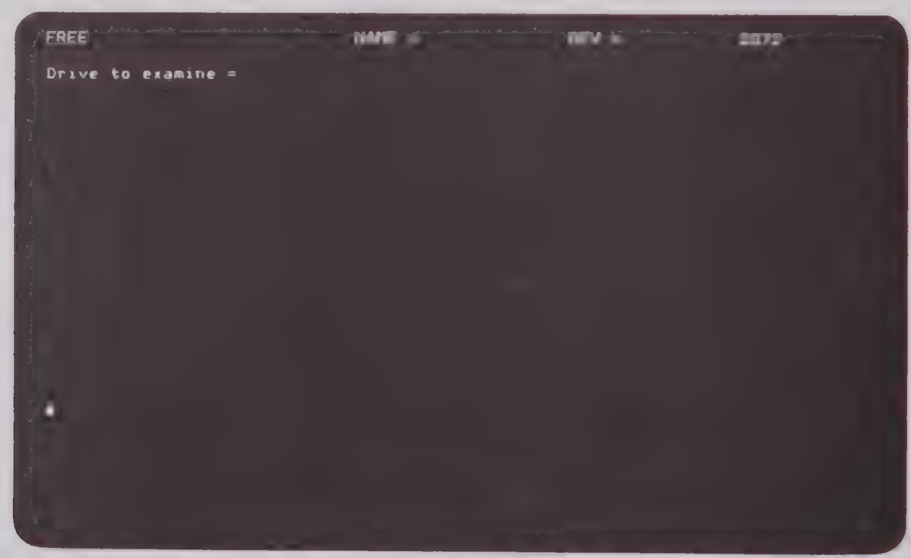
Example

```
#DIR *.*>TI:
```

A directory of the latest versions of all files on the device specified by "DEV =" appears on the terminal screen.

3.5.7 Free Utility

The Free utility determines and displays the number of free blocks (unused space) on a disk cartridge or floppy diskette. The disk drive containing the disk or diskette to be examined for free space is specified and the utility is run. The resulting report is displayed on the video screen. The Free utility options page is shown in Figure 3-11.



```
Drive to examine =
```

Figure 3-11. Free Utility Options Page.

Drive to examine =

Specifies the device containing the disk to be examined for unused space. If no entry is made to this option the device in the "DEV =" option is assumed.

Example

```
#FR SC:
```

Information about the storage space remaining on SC: appears on the screen.

3.5.8 Leditor Utility

The Leditor utility is a "Look only" version of the Editor utility. It allows files to be examined, using standard editor commands to move through the text, but no changes can be made to the file. Searches can be used to locate specific words or characters quickly. The Leditor page (Figure 3-12) is similar to the Editor page except that it does not provide an output file option.

```
LEDITOR                    NAME =           DEV =          2272

Mode [ED2,TECO]

Input =
```

Figure 3-12. Leditor Utility Options Page.

Mode [ED2,TECO]

When the Editor is called from the Background terminal (VT10X) on a Foreground/Background equipped system, this mode option is displayed to allow the user a choice of editors. When the Editor is called from the foreground terminal, this option is not displayed. The version of TECO that can be used with the foreground terminal is automatically called.

[ED2] Enables the ED2 editor (used with VT10X background terminals only).

[TECO] Enables the version of the TECO editor that can be used with the VT10X background terminal.

Input =

The required entry to this option is the extension of the file to be viewed. The "DEV =" and "NAME =" option-values from the top of the page will be used if no alternate device or name is specified.

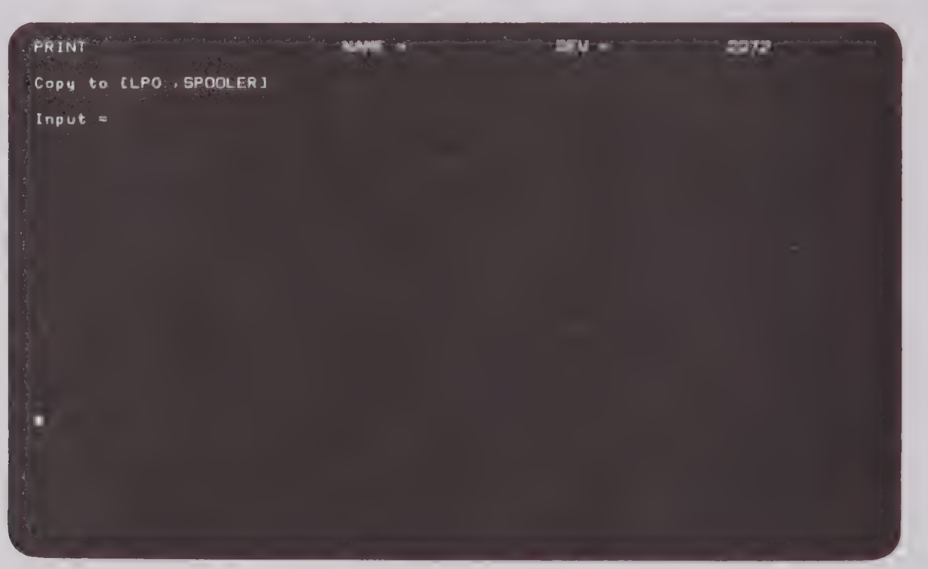
Example

```
#LE SAMPLE.TPG
```

The first page of the file is displayed, and the file may be examined using normal editor commands. Refer to Section 11 for Editor commands. No output file will be created.

3.5.9 Print Utility

The Print utility provides a way of producing a printed (hard-copy) output of text files. These text files can be source program files, report files, listing files, or any ASCII character file. Use the Print utility to specify the file to be printed and the output device on which it is to be printed. The Print utility page is shown in Figure 3-13.



```
PRINT                   NAME =              DEV =           2272

Copy to [LP0:,SPOOLER]

Input =
```

Figure 3-13. Print Utility Options Page.

Copy to [LP0:,SPOOLER]

[LP0:] Selects the character printer. This is the default output device for this utility.

[SPOOLER] Selects the optional line printer and queues additional requests for printing.

Input =

The minimum required entry is the extension of the file to print. An alternate device and/or file name can be specified here, allowing the file to be printed without changing the default "NAME =" and "DEV =" options.

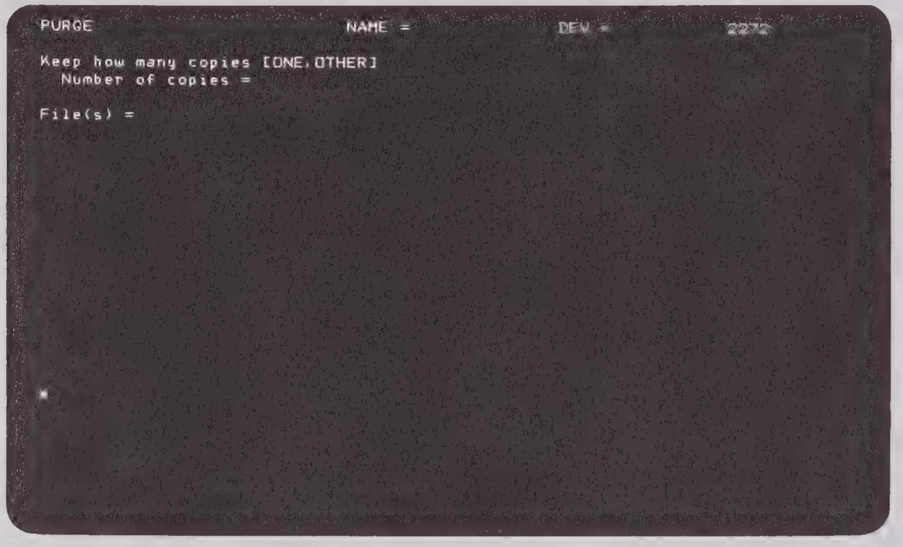
Example

```
#PR SAMPLE.TPG/COPY=SPOOL
```

A command line to spool a file to the line printer.

3.5.10 Purge Utility

The Purge utility is used to purge (remove) old versions of any set of files. The Purge Utility options page is shown in Figure 3-14.



```
PURGE                        NAME =                    DEV =

Keep how many copies [ONE,OTHER]
  Number of copies =

File(s) =
```

Figure 3-14. Purge Utility Options Page.

Keep how many copies [ONE,OTHER]

[ONE] The default setting keeps only one version, the most recent one, of the files specified in the "File(s) =" file spec option.

[OTHER] Selecting OTHER indicates the user wishes to keep more than just the most recent version of the files to purge. Enabling this selection causes the "Number of copies =" option to appear.

Number of copies =

When the OTHER option is selected this option appears on the page. The number of back versions to be kept is entered at this option.

File(s) =

The file specifications of the files to be purged are entered to this option. The minimum required entry is the extension name of the files. If no file-name or disk-device is included in this file specification option, it defaults to the "NAME =" or "DEV =" option at the top of the page. Multiple file specifications, separated by commas, are allowed.

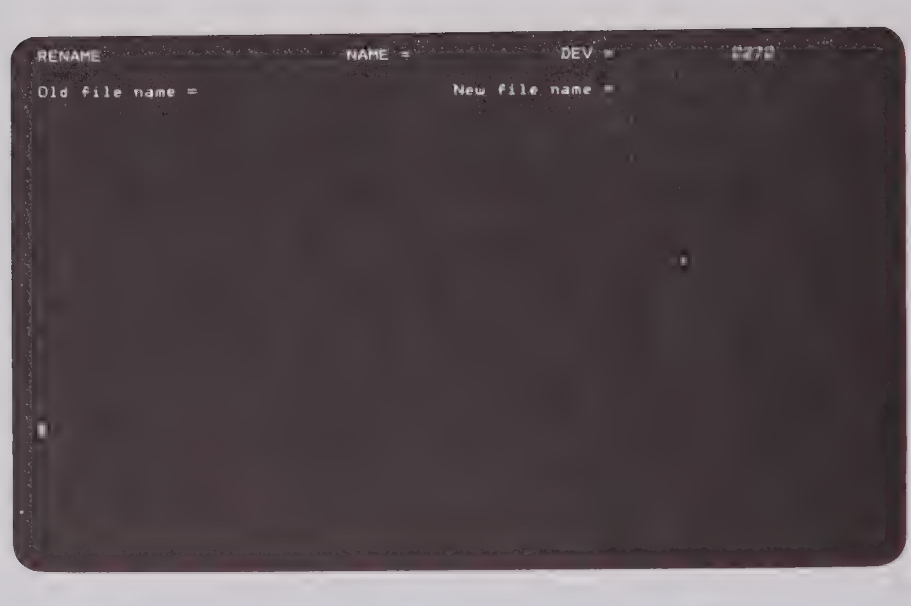
Example

```
#PU SAMPLE.CKT
```

Purges to the latest version of the file, which then will be the only version of that file remaining on the disk.

3.5.11 Rename Utility

Rename allows the name of a file on a disk device to be changed without another version or copy of the file being created. The Rename utility page is shown in Figure 3-15.



```
RENAME                    NAME =               DEV =           2272
Old file name =                        New file name =
```

Figure 3-15. Rename Utility Options Page.

Old file name =
The minimum entry necessary for this option is the extension of the file to be renamed. If no alternate file-name or device is included in this file spec option, it defaults to the "NAME =" or "DEV =" option. If an alternate device is specified, the same device must also be included in the "New file name =" option.

New file name =
The minimum entry to this file spec option is the extension for the new file. The disk-device and file-name default to the "DEV =" and "NAME =" options unless an alternate device and/or file-name appears in this file specification. If a disk-device is included, the same device must also be included in the "Old file name =" option.

Example

```
#REN OLD.TPG>NEW.TPG
```

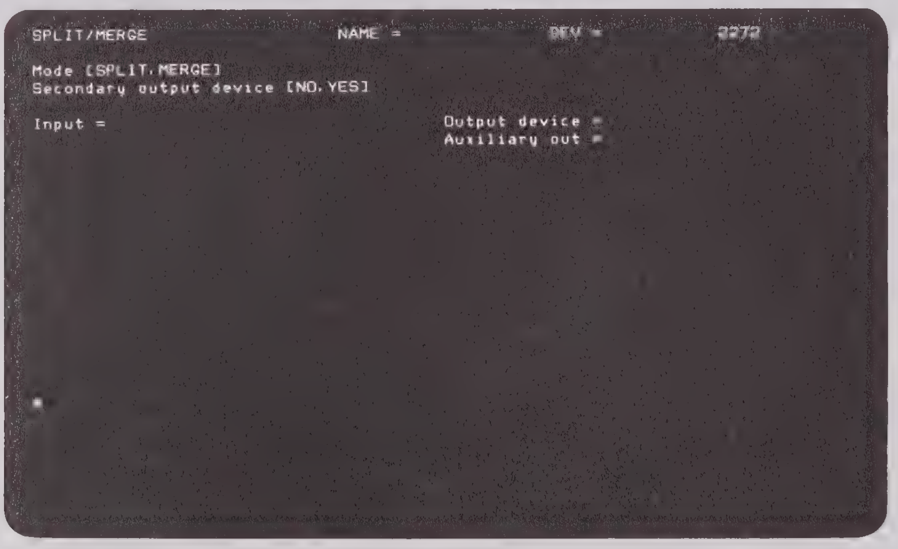
3.5.12 Split/Merge Utility

The Split/Merge utility page, shown in Figure 3-16, provides a means whereby a file from a larger capacity hard-disk cartridge can be copied to a series of smaller disks or diskettes for archiving the file or moving the file to another system.

Before Split mode is run, the disks or diskettes on which the file is to be written must be formatted. Use the Dformat utility to perform this task. Before Merge is run, use the Free utility to determine if enough space (in blocks) is available to permit the file to be copied to that disk. Use the Purge utility to remove (purge) old versions of files on the disk. All versions of an unneeded file can be removed with the Delete utility. Purging and deleting old files increase the free space available on a disk.

The Split mode of this utility transfers a file from the larger disk to a series of smaller disk volumes. For example, a 7000 block file from a Phoenix disk cartridge (DMn: = 27,126 block capacity) in a 2271 or 2272 system could be backed up on a series of 8 floppy diskettes (DYn: = 900 block capacity). To split a file, insert the first formatted disk into the output device, select SPLIT mode, enter the input file specification and output device(s), and run the utility. Split will fill each of the output disks to capacity, if necessary, splitting a contiguous input file between consecutively numbered output disks. Volume numbers, beginning with 1, are automatically assigned to the output disks by the system, and when the next disk or diskette is inserted it is assigned the next higher number. Label each volume with its assigned number so that the disks can be inserted in the proper sequence when the file is later merged. Each output disk is automatically initialized (the directory is cleared) by the split function. Do not attempt to add to a partially filled disk using the Split mode because existing data will be lost.

The Merge mode reassembles the file from the series of smaller disks back into the original form on a larger capacity disk cartridge. When a file is merged on the larger disk, all attributes of the original file, including the UIC, file name, extension and any privilege settings are retained and cannot be changed using the Split/Merge utility. To merge a file, begin by inserting the first volume of the smaller disk series into the input device, select MERGE mode, enter the input file specification and output device, and run the page. The system will request each volume of the series by number, as needed.



```
SPLIT/MERGE                    NAME =                DEV =        3272

Mode [SPLIT,MERGE]
Secondary output device [NO,YES]

Input =                                    Output device =
                                           Auxiliary out =
```

Figure 3-16. Split/Merge Utility Options Page.

Mode [SPLIT,MERGE]

[SPLIT] Selects the Split function.

[MERGE] Selects the Merge function.

Secondary output device [NO,YES]

This option only appears when SPLIT mode is selected.

[NO] Indicates only one surface (e.g., DK2:) or floppy diskette (e.g., DY0:) is to be written to and filled.

[YES] Selecting YES enables the Split mode to write to and fill 2 surfaces or 2 floppy diskettes before new disk(s) must be inserted.

Input =
The minimum required entry is the extension name of the file to be split or merged. If not included in this option, the input device and file name default to the "DEV =" and "NAME =" entries from the top of the page.

Output device =
Specifies the primary disk device to which the output file will be written by either the Split or Merge function.

Auxiliary out =
This option appears when the "Secondary output device" option is set to YES to allow output to two devices by the same Split run. When a multi-surfaced disk or a dual-floppy drive is used as the output device, the second disk surface (e.g., DK3:) or the second floppy (DY1:) can be filled with the transferred input file before another disk(s) must be inserted. This option is not used in Merge mode because, since the input file for Split must come from one disk surface, when the file is merged back to a larger disk it can be written to only one surface.

3.5.13 Type Utility

The Type utility provides a means of displaying text files (not binary files) on the video display. Output from the Type utility can be viewed in screen-sized pages or scrolled continuously from beginning to end. Figure 3-17 shows the Type options page.

The file to be viewed is identified by the "Input =" file specification and, when the utility is run, the first page of the file appears on the video display and remains there. The message --MORE-- near the bottom of the screen indicates that more of the file remains to be seen. The rest of the file can be viewed in screen-sized pages, stopping after each page, or it can be listed continuously at a speed selected by the user. To leave the Type utility before the entire file has been viewed, enter CTRL C (^C) and the monitor page will appear.

To page through the file, press the spacebar and the next page is written to the screen. This action is repeated until the End-Of-File marker appears on the screen. Pressing the spacebar once again causes control to return to either the Type options page or to the monitor page from which Type was called.

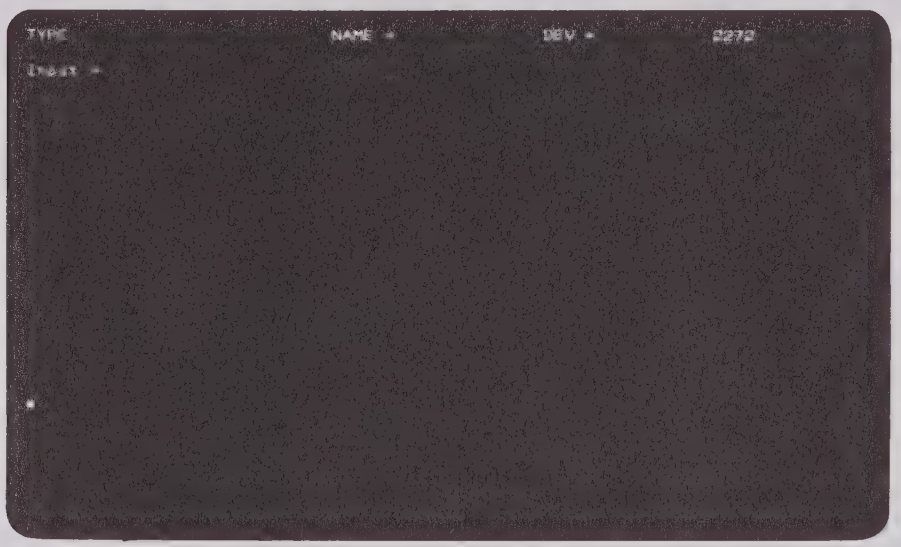
To scroll continuously through the file, press the numeral key described in Table 3-6 to select the display speed and to start the display. To change the display speed while the text is scrolling, press the speed selection key.

Table 3-6
SCROLL CONTROL KEYS ON KEYBOARD

Key	Action
1	Selects FASTEST display speed.
2	Selects SLOWER display speed.
3	Selects even SLOWER speed.
4	Selects SLOWEST display speed.
5-9	Displays only one more line and stops.
0	Turns off continuous scrolling and re-enables the paged display.

Pressing the spacebar while text is scrolling stops the display; otherwise, it restarts the scrolling, as set by the numeral keys previously described. When the End-Of-File marker appears, pressing the spacebar returns to the monitor options page.

Pressing any key other than those described will stop the display.



```
TYPE                    NAME =        DEV =        2272
Input =
```

Figure 3-17. Type Utility Options Page.

Input =

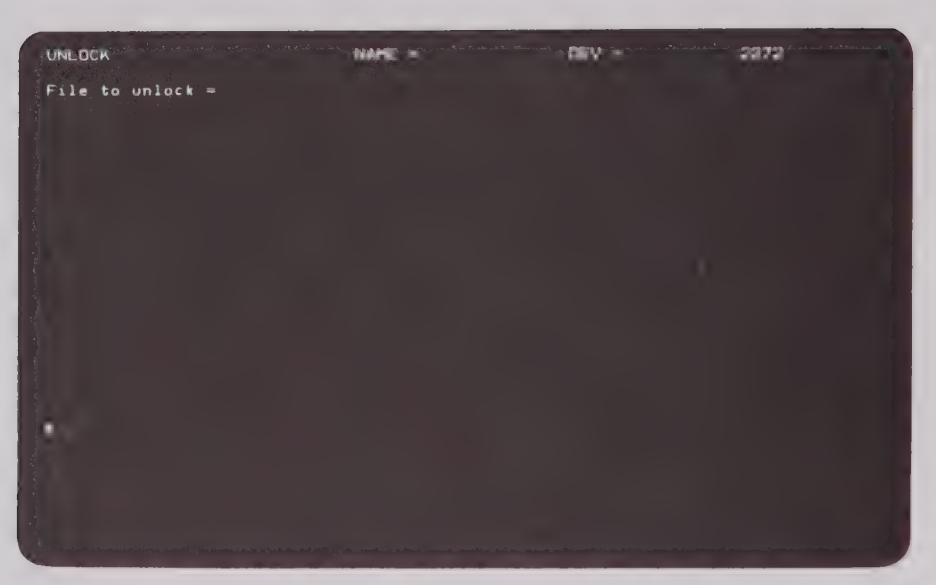
The extension of the file to be viewed is required on this line. The device and file name default to the "DEV =" and "NAME =" entries from the top of the page, unless another device and/or name is included in this file spec.

Example:

```
#TY .TPG
```

3.5.14 Unlock Utility

Disk-files which were improperly closed (e.g., open for write during a power failure) are often "locked" by the operating system. Locked files (identified by the letter "L" in a directory listing) cannot be opened for read or write access. The Unlock utility unlocks improperly closed files so they can be accessed again. Sometimes a file is corrupted by whatever caused it to be improperly closed. Use caution when proceeding to use an unlocked file. The Unlocked utility page is shown in Figure 3-18.



```
UNLOCK                  NAME =             DEV =         [illegible]
File to unlock =
```

Figure 3-18. Unlock Utility Options Page.

File to unlock =

The extension name of the locked file is the minimum required entry to this file spec option. A disk-device and/or file-name may be included to override the "NAME =" and "DEV =" default values.

<u>Example</u>

```
#UN .CKT
```

An improperly closed file (with the extension .CKT) is unlocked and made available again. The "L" disappears from the directory listing.

Test Set Analysis and Special Considerations – Section 4

4.1 GENERAL

When a new pc board design is released to manufacturing, the manufacturing test personnel must analyze the design and formulate a plan for testing the pc board during the manufacturing process. A subset of the overall testing plan includes the design of a test program, a test fixture, and related instructions for a test operator. These items are referred to as a "test set." Some of the areas of concern to the test personnel in designing a test set are:

What information and materials are needed?

What components can readily be tested by the 227X and which need special routines written?

Does the 227X configuration provide adequate power supply and other options?

What decisions are to be made concerning fixture design and procurement?

What form of data collection should be used?

What tasks must be performed to create the test set?

Section 1 of this manual discussed much of the theory of testing on a 227X tester, Section 2 discussed how to operate the 227X tester, and Section 3 discussed file formats and file management. An understanding of these 3 subjects is necessary before the various steps and decisions for creating the test set can be effectively accomplished. This section provides some insight on how to make decisions and on how to proceed with the design of a test set. The methods presented are not necessarily the only, or even the best method for each situation. They are only presented as a guide for a new 227X user.

4.2 INTRODUCTION OF TUTORIAL CIRCUIT

A circuit has been created for the purpose of illustrating the various procedures discussed in this manual. It is a circuit that provides a broad coverage of devices and situations which are of concern to users as they prepare test sets. A pc board (called Customer Training Board) of this circuit has been fabricated for the 2275 system and is available from GenRad. With this pc board the user can actually try examples presented in the manual. This board and a test fixture, illustrated in Figure 4-1, are provided with each 2275 system. The pc board could be used on 2271 and 2272 systems by using the 2272 fixture and the appropriate fixture converter.

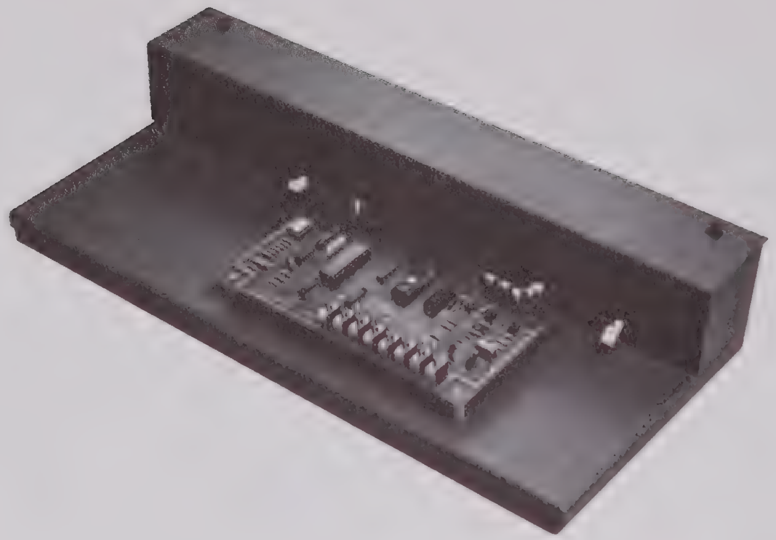


Figure 4-1. Training Board and Fixture for 2275.

4.3 OUTLINE OF STEPS IN TEST SET PREPARATION

Preparation of a test set for a circuit board requires these activities:

Gather and analyze board data and prepare a test plan. Refer to para 4.4.
This is the informal activity that one performs to understand the circuit to be tested and to formulate an approach to design a test.

Prepare input files for special test requirements. Refer to para 4.5.
This includes those activities which involve preparation of special file elements to deal with special (unusual) components and circuitry on the pc board.

Mark test nodes on schematic. Refer to para 4.5.6.
This activity is recommended as the formal starting point for preparation of all pc board test sets.

Create a circuit description file using CKTGEN. Refer to Section 6.
This activity creates a basic user input file that describes the pc board. The activity includes responding to a menu prompt to enter the descriptive information.

Generate a source test program using ATG. Refer to Section 7.
This activity involves the running of the various phases of the automatic test generation process. It uses the circuit description file and creates a test program.

Generate a fixture wiring list using NAIL ASSIGNMENT or FIXTURE REPORT. Refer to Section 8.
This activity involves the running of the process that automatically assigns test nails and provides wiring lists for the test fixture.

Design and build the test fixture. Refer to Test Fixture Kit manual.
This activity involves those things that are required in order to obtain a test fixture.

Convert the source program to an object code test file using TRANSLATE, DEBUG and UPDATE. Refer to Section 9.
These activities include the creating of a machine-readable test program and using the facilities available to the programmer for debugging the test set and updating the test set with changes made during debug.

Provide operator test information. Refer to Section 10.
This activity should result in information that the production test operator needs to know about the test set in order to conduct testing.

The flow of these activitiies and the primary files used and created by them are illustrated in Figure 4-2.

At this point it may be helpful to review the differences between 227X Board Test Workstations as they pertain to tasks and files used in test set design. The 2270 has one scanner output and/or one driver/sensor for each test point. The 2271/72/75 has a multiplexing arrangement which allows each scanner output and driver/sensor to be switched to several test points. This means that test points and consequently test fixture nails cannot be assigned arbitrarily by the user on the 2271/72/75 systems.

The software automatically assigns test points in accordance with system multiplexing systems. The user assigns sequential node numbers so that temporary files can be created for the ATG process. The Nails Assignment task later assigns real numbers and updates all temporary files with these nail numbers. The updated files are given a new extension. The temporary and updated files involved in the process are shown in Figure 4-3.

When the target system is a 2270, temporary files are not created and Nail Assignment is not performed. The discussion in this section may use the file extensions interchangeably so the reader should keep in mind which extension is correct for the system being used.

4.4 THE TEST PLAN

4.4.1 General

A test plan should be prepared to document the requirements for a good test set. The plan cannot be written until information is obtained. The information needed is a schematic, parts data, and engineering data.

Review the schematic diagram to determine what special testing requirements exist for the circuit. These requirements could include components not modeled in any library, functional tests for unique circuit segments, or extra power-supply voltage or current needs. Decisions then must be made as to how to program the special test requirements. These decisions must be outlined because they will affect both the preparation of the circuit description and the overall schedule of preparing the test set. The decisions may also effect system option hardware needs. Figure 4-4 illustrates the decision process that may be used in deciding how to test the board. Figure 4-5 illustrates the test plan process.

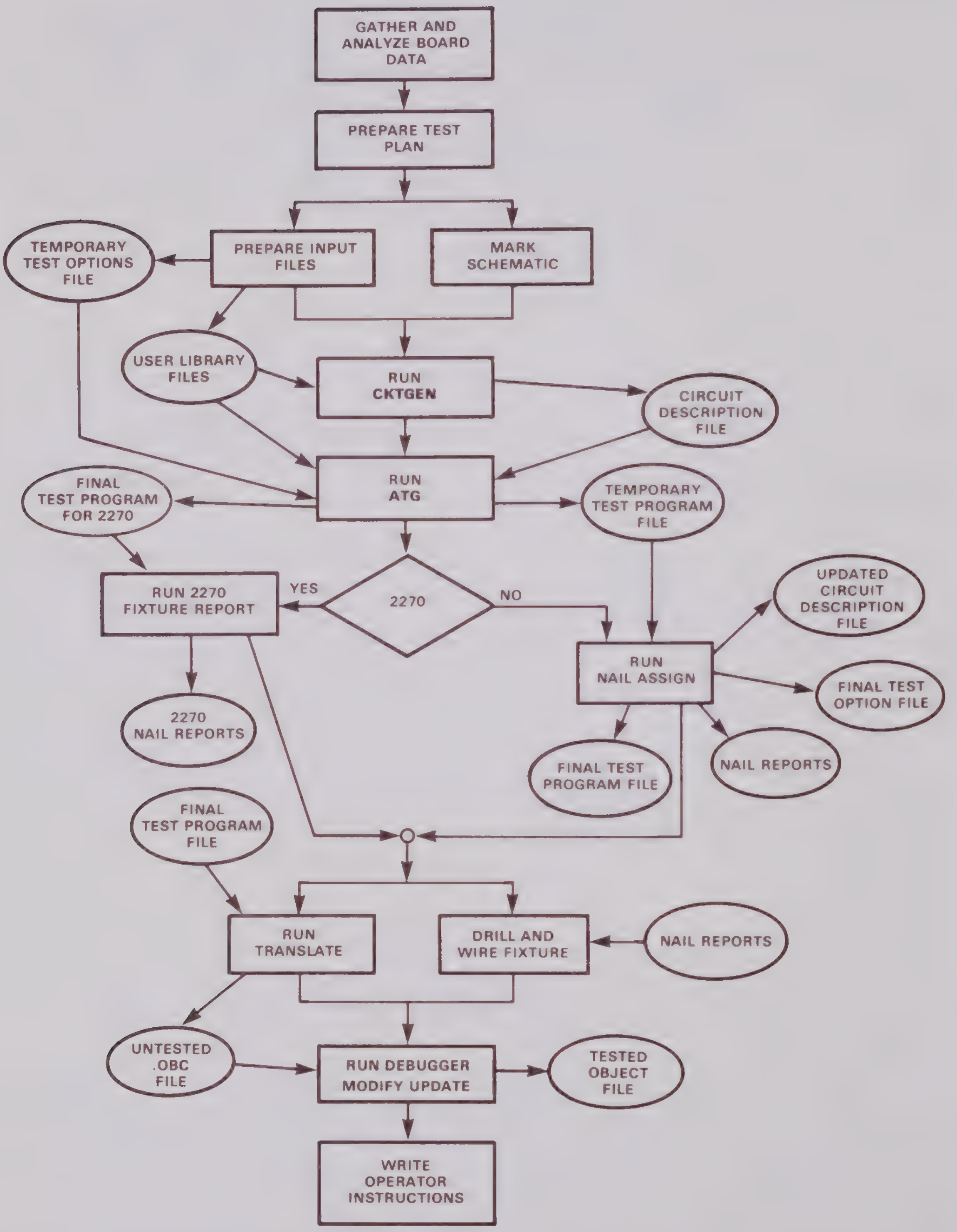

GATHER AND ANALYZE BOARD DATA
PREPARE TEST PLAN
TEMPORARY TEST OPTIONS FILE
PREPARE INPUT FILES
MARK SCHEMATIC
USER LIBRARY FILES
RUN CKTGEN
CIRCUIT DESCRIPTION FILE
FINAL TEST PROGRAM FOR 2270
RUN ATG
TEMPORARY TEST PROGRAM FILE
RUN 2270 FIXTURE REPORT
YES
2270
NO
UPDATED CIRCUIT DESCRIPTION FILE
RUN NAIL ASSIGN
FINAL TEST OPTION FILE
2270 NAIL REPORTS
FINAL TEST PROGRAM FILE
NAIL REPORTS
FINAL TEST PROGRAM FILE
RUN TRANSLATE
DRILL AND WIRE FIXTURE
NAIL REPORTS
UNTESTED .OBC FILE
RUN DEBUGGER MODIFY UPDATE
TESTED OBJECT FILE
WRITE OPERATOR INSTRUCTIONS

Figure 4-2. Primary Activities and Files of the Test Set Preparation Process.

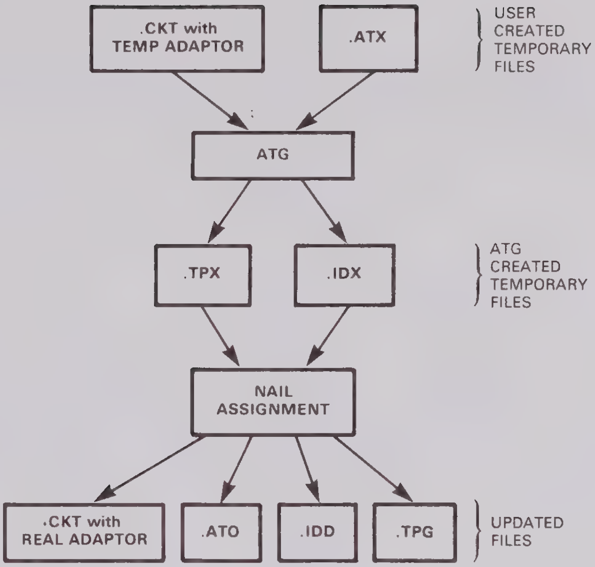

.CKT with TEMP ADAPTOR
.ATX
USER CREATED TEMPORARY FILES
ATG
.TPX
.IDX
ATG CREATED TEMPORARY FILES
NAIL ASSIGNMENT
.CKT with REAL ADAPTOR
.ATO
.IDD
.TPG
UPDATED FILES

Figure 4-3.
Files on Multiplexed Systems.

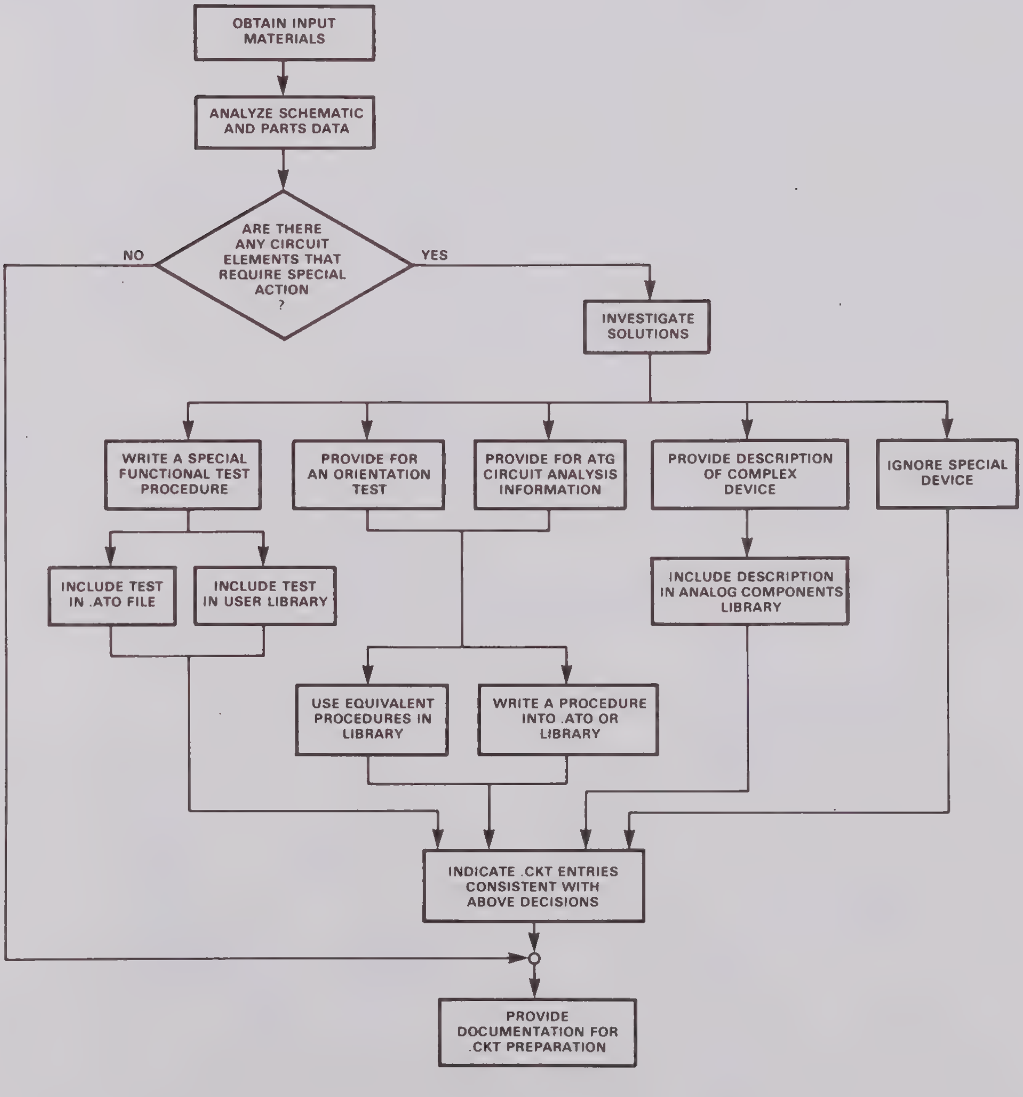

OBTAIN INPUT MATERIALS
ANALYZE SCHEMATIC AND PARTS DATA
ARE THERE ANY CIRCUIT ELEMENTS THAT REQUIRE SPECIAL ACTION ?
NO
YES
INVESTIGATE SOLUTIONS
WRITE A SPECIAL FUNCTIONAL TEST PROCEDURE
PROVIDE FOR AN ORIENTATION TEST
PROVIDE FOR ATG CIRCUIT ANALYSIS INFORMATION
PROVIDE DESCRIPTION OF COMPLEX DEVICE
IGNORE SPECIAL DEVICE
INCLUDE TEST IN .ATO FILE
INCLUDE TEST IN USER LIBRARY
INCLUDE DESCRIPTION IN ANALOG COMPONENTS LIBRARY
USE EQUIVALENT PROCEDURES IN LIBRARY
WRITE A PROCEDURE INTO .ATO OR LIBRARY
INDICATE .CKT ENTRIES CONSISTENT WITH ABOVE DECISIONS
PROVIDE DOCUMENTATION FOR .CKT PREPARATION

Figure 4-4. Typical Decision Process Used in Test Set Planning.

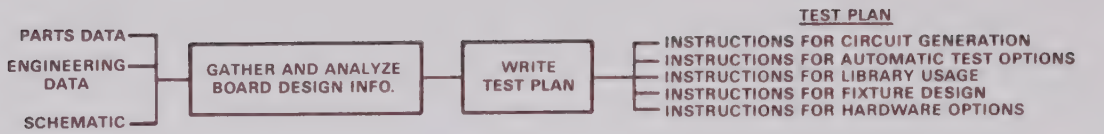

PARTS DATA
ENGINEERING DATA
SCHEMATIC
GATHER AND ANALYZE BOARD DESIGN INFO.
WRITE TEST PLAN
TEST PLAN
INSTRUCTIONS FOR CIRCUIT GENERATION
INSTRUCTIONS FOR AUTOMATIC TEST OPTIONS
INSTRUCTIONS FOR LIBRARY USAGE
INSTRUCTIONS FOR FIXTURE DESIGN
INSTRUCTIONS FOR HARDWARE OPTIONS

Figure 4-5. Test Planning Process.

4.4.2 Breadboarding Solutions

An important function in the decision process could be investigation of possible solutions. Breadboarded circuits could be built for investigative testing. A simple fixture would be needed to plug this circuit into. A universal fixture could be made for this purpose. The 2272 test system makes it easy by providing a connector on the OVP box so that a fixture is not needed. The connector on the OVP box provides access to selected test points.

Tests can be designed to use these test points in the same way as the nails on a fixture are used, except that a circuit board is connected to the test points through the OVP box. See Figure 4-6. The 5-volt supply is hardwired to the breadboard. Thus, shorts tests should not be included in any breadboard test program. If analog ATG is run, a shorts test is inserted in the test program.

A circuit board is provided, with each 2272 OVP box, which can be used for breadboarding. Additional boards may be ordered by Part No. 2272-4723. Users may also use boards of their own that are compatible with the 80-pin pc connector. The circuit board provided is used by mounting components on the board and wirewrapping the interconnections and the node connections to the appropriate test points on the edge connector. Table 4-1 shows which system test points are connected to each edge connector pin.

A test fixture and modeling breadboard are provided with each 2275 Workstation. The fixture is the same one used for the user training board. Dual 50-pin connectors are used. The wiring of these connectors to the fixture interface boards is shown in Table 4-2.

A test procedure can be developed for an IC by wiring it on the breadboard and trying test procedures under system control until a good test is achieved. Figure 4-7 illustrates the wiring of an IC to the breadboard.

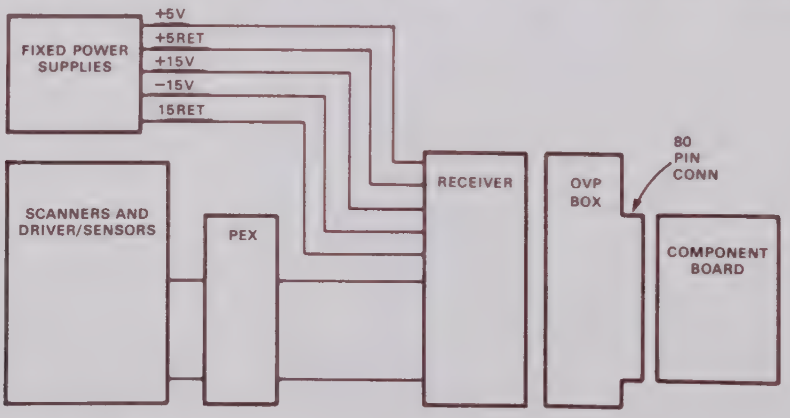
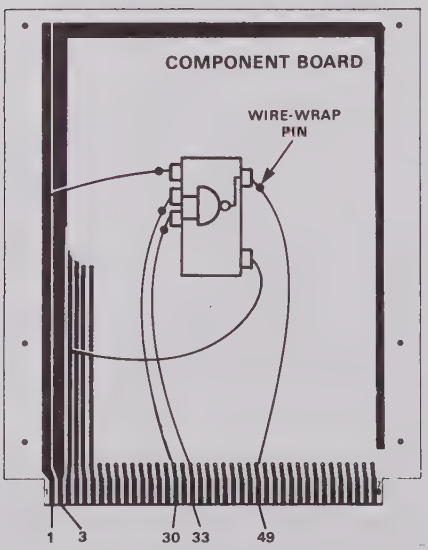

FIXED POWER SUPPLIES
+5V
+5RET
+15V
−15V
15RET
SCANNERS AND DRIVER/SENSORS
PEX
RECEIVER
OVP BOX
80 PIN CONN
COMPONENT BOARD

Figure 4-6. Relationship of 2272 Breadboard to Test System.

Table 4-1
2272 BREADBOARD CONNECTIONS

Breadboard Pin	System Test Points	Breadboard Pin	System Test Point
1	5 V HI	41	370,378
2	5 V HIS	42	GND
3	5 V LO	43	385,393
4	5 V LOS	44	401,409
5	+15 V	45	417,425
6	+15 V RET	46	433,441
7	-15 V	47	449,457
8	GND	48	465,473
9	129,137	49	481,489
10	145,153	50	497,505
11	161,169	51	386,394
12	177,185	52	402,410
13	193,201	53	418,426
14	209,217	54	434,442
15	225,233	55	450,458
16	241,249	56	466,474
17	130,138	57	482,490
18	146,154	58	498,506
19	162,170	59	GND
20	178,186	60	513,521
21	194,202	61	529,537
22	210,218	62	545,553
23	226,234	63	561,569
24	242,250	64	577,585
25	GND	65	593,601
26	257,265	66	609,617
27	273,281	67	625,633
28	289,297	68	514,522
29	273,281	69	625,633
30	321,329	70	546,554
31	337,345	71	562,570
32	353,361	72	578,586
33	369,377	73	594,602
34	258,266	74	610,618
35	272,282	75	626,634
36	290,298	76	No Connection
37	306,314	77	No Connection
38	322,330	78	No Connection
39	338,346	79	No Connection
40	354,362	80	No Connection



```
ASSIGN LGC VIHA = 4.5 VILA = 0.0 VOHA = 2.4 VOLA = 0.8
LVLA(321, 369, 481)
IC(321, 369) IL(321, 369) OS(481) OH(481)
```

Figure 4-7. Example of Breadboard Technique.

Table 4-2
2275 BREADBOARD CONNECTIONS

Breadboard Pin	Fixture Interface Board and Pin	System Test Point	Breadboard Pin	Fixture Interface Board and Pin	System Test Point
P11-1	5-22	146	P12-1	0-49	+5 V
2	5-23	147	2	0-51	+5 V
3	5-24	148	3	0-53	+5 V
4	5-25	149	4	0-55	GND
5	5-26	150	5	0-57	GND
6	6-17	173	6	0-59	GND
7	6-18	174	7	0-63	+15 V
8	6-19	175	8	0-67	GND
9	6-20	176	9	0-69	-15 V
10	6-21	177	10	1-17	13
11	6-22	178	11	1-18	14
12	6-23	179	12	1-23	19
13	6-24	180	13	1-24	20
14	6-25	181	14	1-25	21
15	7-17	205	15	1-26	22
16	7-18	206	16	2-17	45
17	7-19	207	17	2-18	46
18	7-20	208	18	2-19	47
19	7-21	209	19	2-20	48
20	7-22	210	20	2-21	49
21	7-23	211	21	2-22	50
22	7-24	212	22	2-23	51
23	7-25	213	23	2-24	52
24	8-17	237	24	2-25	53
25	8-18	238	25	2-26	54
26	8-19	239	26	3-17	77
27	8-20	240	27	3-18	78
28	8-21	241	28	3-19	79
29	8-22	242	29	3-20	80
30	8-23	243	30	3-21	81
31	8-24	244	31	3-22	82
32	8-25	245	32	3-23	83
33	9-17	269	33	3-24	84
34	9-18	270	34	3-25	85
35	9-19	271	35	3-26	86
36	9-20	272	36	4-17	109
37	9-21	273	37	4-18	110
38	9-22	274	38	4-19	111
39	9-23	275	39	4-20	112
40	9-24	276	40	4-21	113
41	9-25	277	41	4-22	114
42	10-17	301	42	4-23	115
43	10-18	302	43	4-24	116
44	10-19	303	44	4-25	117
45	10-20	304	45	4-26	118
46	10-21	305	46	5-17	141
47	10-22	306	47	5-18	142
48	10-23	307	48	5-19	143
49	10-24	308	49	5-20	144
50	10-25	309	50	5-21	145

4.5 USER-PREPARED INPUT FILES

Once a test plan is made, the test set preparers start preparing test software. There are several files that the preparers may need to generate and customize the test set to the pc board. Instructions for preparing the circuit description file should also be written. Figure 4-8 summarizes the process of preparing user-generated files for customizing a test set.

4.5.1 Write a Test Procedure

This decision may be chosen when a digital IC is on the pc board and is not modeled in the digital libraries. The best approach for including the special test in the .TPG is to insert the test in the User Digital Library so that ATG can include data about the nodes of the component in the In-circuit Diagnostic Data file (.IDD) and so that ATG can know more about the circuit for its circuit analysis function. Information for writing library routines and for inserting these routines into the user library files is provided in the 227X

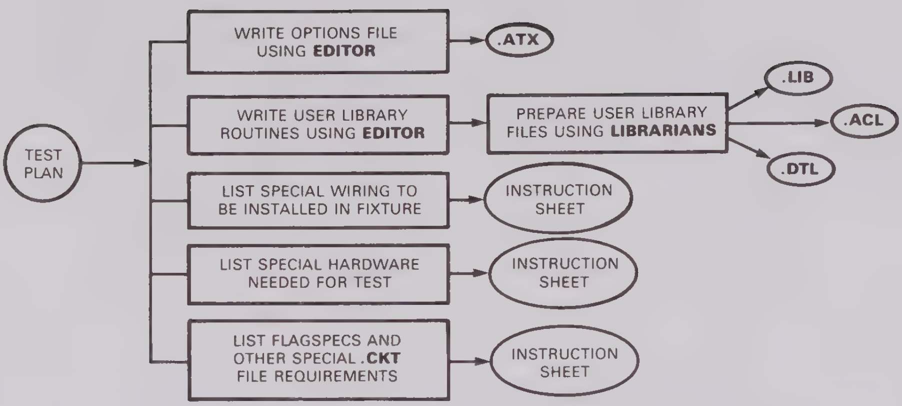

TEST PLAN
WRITE OPTIONS FILE USING **EDITOR**
.ATX
WRITE USER LIBRARY ROUTINES USING **EDITOR**
PREPARE USER LIBRARY FILES USING **LIBRARIANS**
.LIB
.ACL
.DTL
LIST SPECIAL WIRING TO BE INSTALLED IN FIXTURE
INSTRUCTION SHEET
LIST SPECIAL HARDWARE NEEDED FOR TEST
INSTRUCTION SHEET
LIST FLAGSPECS AND OTHER SPECIAL .CKT FILE REQUIREMENTS
INSTRUCTION SHEET

Figure 4-8. User-Prepared Input for Customizing a Test Set.

Test Library Programming manual. When this decision is implemented, the .CKT entry is similar to that of any IC type that is included in a library.

An alternate method is to write a test routine for the IC and to include the routine in the .ATO file. ATG will then automatically replace a dummy routine, which would otherwise have been taken from the library, with the routine in the .ATO file. This method does not have the advantages of complete .IDD data with disable, inhibit, and force information. Another disadvantage is that, if the component is used on more than the one board type, the routine would have to be included in each board's .ATO file rather than just once in the library. The .CKT entry for this method would be of the DICnn form. Information for writing test routines is in the 227X Test Programming manual. Section 5 of this manual describes how to use .ATO files.

The third, and least desirable, method of writing a test procedure is to edit the .TPX file. This method does not include complete .IDD data and the .TPX would have to be edited every time ATG was run to generate a .TPX file. In this case, the .CKT entry should also be of the form, DICnn.

4.5.2 Provide for an Orientation Test

It may not be desirable to spend the programming effort to provide a functional test for an IC, but a test that would determine proper insertion into the pc board would be desirable. This may be accomplished just by treating two terminals of the device as a simple component, such as a resistor, diode, or capacitor. In this case, only an entry in the .CKT file is required. If the situation is more complicated, the desired result may be accomplished by inserting a device description into the user analog component library. Information on analog component library usage is provided in the 227X Test Library Programming manual.

4.5.3 Provide for Circuit Analysis Information

A component may exist for which the decision is not to provide any test. It is necessary, however, to provide enough information so that some circuit analysis can be accomplished by ATG and a complete .IDD file can be made. This may be important for a digital IC where accurate disable, inhibit, and force information is required to test effectively other ICs in the circuit. In this case, an entry can be made in the user digital library for the device. This entry should include all parts of a library routine; the .HEAD, .DISABLE, .INHIBIT, .LFORCE, .HFORCE, and .MAIN, as applicable for disabling, inhibiting and forcing the IC. The main section is a dummy that should use every IC pin. For example:

```
MAIN;

   IC(1-22) OS(23,24);

END MAIN;
```

Include all components in the .CKT to get good circuit analysis. When it is obvious that no test can be generated for a component, such as a small bypass capacitor in parallel with a large capacitor, the circuit connections should be provided so that ATG can test the large capacitor. The NOTEST flagspec should be used with the small capacitor to save ATG time.

4.5.4 Provide Component Description in ACL

A special component may be made up of several simple analog components, such as a resistor pack, or it may be a device that could be modeled as one of the basic analog components. In these cases, a description of the device can be written into the User Analog Component Library so that ATG can automatically generate a test. Device descriptions are written into the .ACL file, using the same general rules that apply to writing a .CKT file, except that there is no Adaptor section in a analog component description. Refer to Section 6 and the 227X Test Library manual for details on writing ACL descriptions.

4.5.5 Ignore Special Device

If it is decided not to test a device or consider its effect in testing other devices, it can essentially be ignored. This can be done by making no reference to its pins or by naming the device and allowing the system to default to an unspecified analog IC. In the latter case, its pins will be represented in the .IDD file and the back-to-back diode model will be used in circuit analysis.

4.5.6 Mark Schematic For Circuit Description File Generation

The basic programming task of a user is creating a circuit description source (.CKT) file for a pc board to be tested. The .CKT file is a formatted listing of circuit component descriptions and circuit interconnections. The CKTGEN monitor routine is generally used to create this file. CKTGEN prompts the user to enter all the required circuit information. The operation of CKTGEN and the description of CDL are contained in Section 6.

A schematic that agrees with the design of the known-good boards must be provided. It should be a film positive for ease of marking and copying.

The programmer should mark each node (common circuit point) with a unique node name (e.g., N1, N2, etc). If these node names have a numerical value, then the node name can be considered as temporary nail numbers which can be easily cross referenced to permanent nail numbers produced by nail assignment. Nail Assignment converts 2271, 2272 or 2275 temporary nail numbers to actual nail numbers in the .TPG. A suggested method of marking for 2271, 2272 and 2275 systems is to write on the schematic diagram, next to each node, the temporary nail number followed by a slash (/) and a space. The space is filled in later with permanent nail numbers.

A parts list should provide the value and tolerance of all component characteristics so that each component can be accurately described in the %VALUE section of the .CKT file. Commercial type numbers of all ICs should be provided so that the proper library procedure can be used or so that a special test can be written if a test is not in the library. Data sheets and circuit descriptions should be provided for custom components to allow a special functional test, an orientation test, or an analog component description to be written.

4.5.7 CAD Post Processors

Most major CAD systems now have ATE post processors available which are capable of producing parts of the GenRad circuit description file. GenRad also provides a networking link (CADMATE) for transferring files from a CAD system to a GenRad test system. Check with your CAD facility and your GenRad representative for more information.

4.6 INSTRUMENTATION

4.6.1 General

When tests for special components are being written, it is likely that power requirements will dictate a need for power sources other than the standard analog source and measure devices and the fixed dc supplies. Several additional supplies are available as system options. The programmer should determine if adequate supplies exist in the system and determine in what configuration they exist. A description of all power-supply options is given in the appropriate system operation manual. The configuration is important so that the appropriate program command statements for controlling a supply can be determined. The configuration also affects the test fixture wiring design. Use the SETUP monitor page described in para 4.9 to determine the configuration of a system. The programming name for each power-supply option is shown in Figure 4-9. The names consist of a supply type and a slot location. Refer to para 4.6.6 for details on programming power supplies.

It is not recommended that these power supplies be connected directly to the circuit. They should be connected through relays so that the supplies can be disconnected when not in use. Relays are provided in optional relay boxes for disconnecting TPS and/or UPS power supplies. HCS power supplies have their own relays. All power-supply relays are controlled by the relay driver board (also called parallel input/output (PIO) board). This PIO board is a standard system item inserted in PIO slot 0 and its programming name is PIO(0).

Each type of power supply is described in the following paragraphs.

4.6.2 Fixed Supplies

The fixed supplies are set for the following outputs:

±15 V, 0.75 A

+5 V, 3 A

The fixed, non-programmable output voltages from each supply are applied whenever 2270/71/72 system power is on. However, a user-supplied relay energized by a relay driver may be used to connect or disconnect this fixed voltage to the

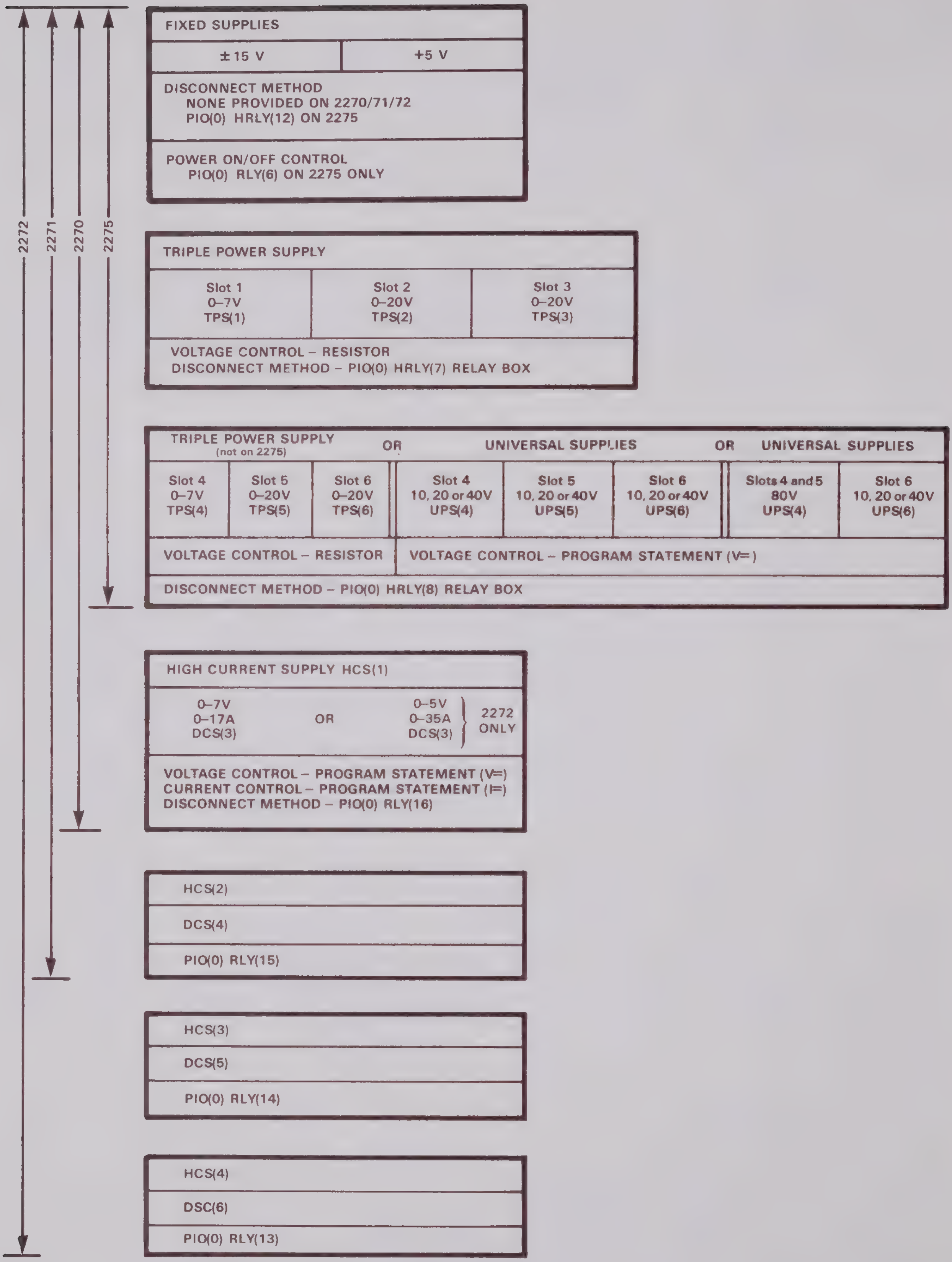

2272
2271
2270
2275
FIXED SUPPLIES
±15 V
+5 V
DISCONNECT METHOD
NONE PROVIDED ON 2270/71/72
PIO(0) HRLY(12) ON 2275
POWER ON/OFF CONTROL
PIO(0) RLY(6) ON 2275 ONLY
TRIPLE POWER SUPPLY
Slot 1 0–7V TPS(1)
Slot 2 0–20V TPS(2)
Slot 3 0–20V TPS(3)
VOLTAGE CONTROL – RESISTOR
DISCONNECT METHOD – PIO(0) HRLY(7) RELAY BOX
TRIPLE POWER SUPPLY (not on 2275)
OR
UNIVERSAL SUPPLIES
OR
UNIVERSAL SUPPLIES
Slot 4 0–7V TPS(4)
Slot 5 0–20V TPS(5)
Slot 6 0–20V TPS(6)
Slot 4 10, 20 or 40V UPS(4)
Slot 5 10, 20 or 40V UPS(5)
Slot 6 10, 20 or 40V UPS(6)
Slots 4 and 5 80V UPS(4)
Slot 6 10, 20 or 40V UPS(6)
VOLTAGE CONTROL – RESISTOR
VOLTAGE CONTROL – PROGRAM STATEMENT (V=)
DISCONNECT METHOD – PIO(0) HRLY(8) RELAY BOX
HIGH CURRENT SUPPLY HCS(1)
0–7V 0–17A DCS(3)
OR
0–5V 0–35A DCS(3)
2272 ONLY
VOLTAGE CONTROL – PROGRAM STATEMENT (V=)
CURRENT CONTROL – PROGRAM STATEMENT (I=)
DISCONNECT METHOD – PIO(0) RLY(16)
HCS(2)
DCS(4)
PIO(0) RLY(15)
HCS(3)
DCS(5)
PIO(0) RLY(14)
HCS(4)
DSC(6)
PIO(0) RLY(13)

Figure 4-9. User Power-Supply Configurations.

unit-under-test. In the 2275, the fixed supplies are switched on by relay driver HRLY(12) on PIO(0). They are connected to the circuit by relay driver RLY(6) on PIO(0).

To ensure proper dc regulation of the 5-V supply, sense leads 5 V HIS and 5 V LOS should be tied to voltage output lines 5 V HI and 5 V LO, respectively, at the external load.

4.6.3 Triple Power Supply (Optional)

The TPS contains one 0-7 V supply and two 0-20 V supplies. The current ratings for these supplies on the 2275 differ from those on the 2270/71/72 systems, as follows:

Voltage (V)	System	Current (A)
0-7	2270/71/72	5
0-7	2275	7.5
0-20	2270/71/72	1.5
0-20	2275	3

The output from each supply is floating and can be wired to provide either a positive or negative voltage to a variety of analog and digital devices. Each power-supply output is independently programmable to any of 3 selectable voltage levels. These voltage levels are determined by programming resistors, as described later in this section.

Output voltage levels for each of the three power supplies in a TPS package are set by programming resistors. On a 2270/71/72 they are mounted on a 24-pin connector[1] that plugs into the rear (standard unit) or front (optional unit) of the supply chassis. On a 2275 they are mounted on an interface board that mounts in the test fixture. Because the output voltage is proportional to the conductance of the programming resistor, it is only necessary to change the resistance value in order to change the output level of each power supply. The programming accuracy of each power supply output is ±0.4% full scale plus the accuracy of the programming resistor. The formula that may be used to determine the proper resistance value for any power supply is:

$$R_{prog} = \frac{100K}{V_{output}} - 45$$

---

[1] GR 2200-9747 or Amphenol 26-4100-24P. A GR 2200-9747 package of five blank programming plugs is also available. Each TPS is shipped with a single plug installed. The plug has ± 1% film resistors that establish the TPS output levels as follows:

P.S. No.	A-Level	B-Level	C-Level
1	4.75 V	5.0 V	5.25 V
2	13.5 V	15.0 V	16.5 V
3	13.5 V	15.0 V	16.5 V

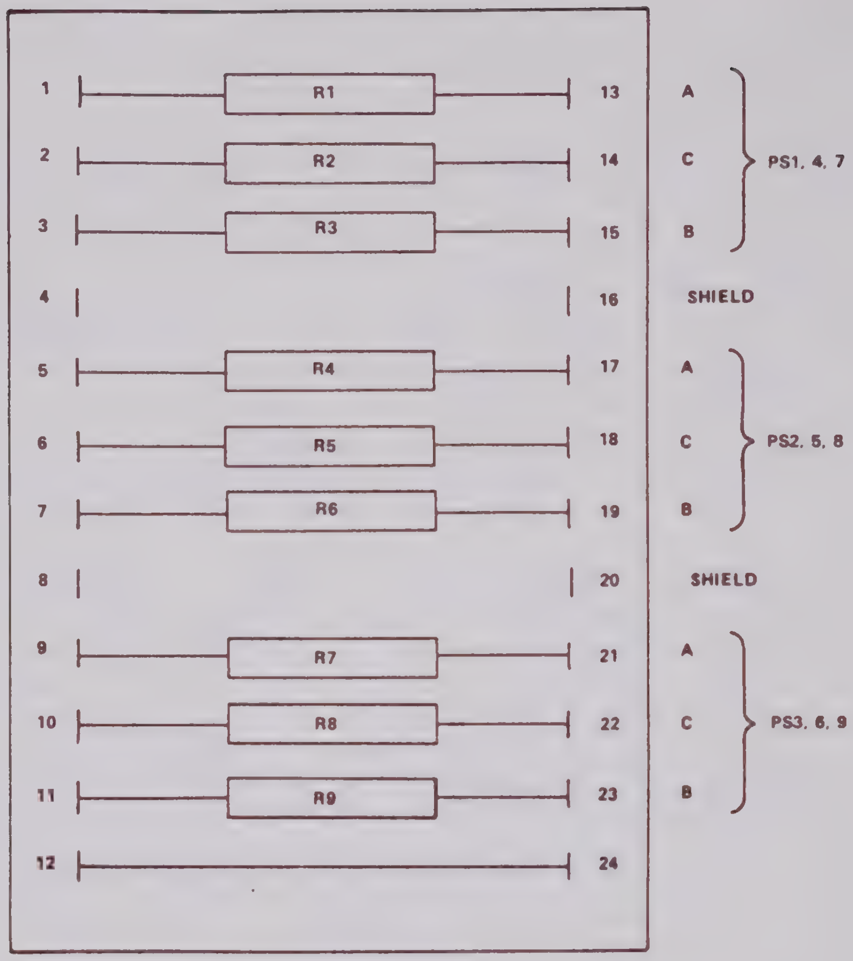

1
R1
13
A
2
R2
14
C
PS1, 4, 7
3
R3
15
B
4
16
SHIELD
5
R4
17
A
6
R5
18
C
PS2, 5, 8
7
R6
19
B
8
20
SHIELD
9
R7
21
A
10
R8
22
C
PS3, 6, 9
11
R9
23
B
12
24

Figure 4-10. TPS Voltage Level Programming.

where:	Rprog	The resistor value in ohms.
	100K	A fixed constant.
	Voutput	Desired output level in V.
	45	The total percent accuracy X 100 (e.g., power supply is 0.4% and RPROG is 0.05%).

For example, if an output level of 5 V is desired for one of the three resistor selectable outputs (A, B, and C) of the 0-7 V TPS, the formula can be used to derive a resistance value of 19,955 ohms.

$$\text{Rprog} = \frac{100{,}000}{5} - 45 = 19{,}955$$

A 19,955-ohm resistor would therefore develop a 5-V output to an accuracy equal to the resistor tolerance plus 0.4%.

In 2270, 2271, and 2272 systems, resistors are soldered to the programmable connector pins shown in Figure 4-10. Drawing 2200-2058 in the Diagrams manual details the connector assembly. It is recommended that a group of blank programmable connectors be obtained so that various output voltage levels can be established in advance of testing by simply plugging in a different connector. Removing and re-soldering different resistors to the same 24-pin connector are not recommended. Be sure to provide the programming plug information to the appropriate personnel so that the plug can be assembled in time for debugging.

The jumper between points 12 and 24 on the PROGRAMMABLE INPUT plug allows the test system to sense both the front and rear plugs. If both plugs are installed, the TPS is forced into the off state and an error message will be generated when the system assurance test is run.

In 2275, systems the TPS programming resistors are soldered to terminals on the fixture interface board 21. Therefore, the correct TPS levels are automatically programmed whenever the test fixture is installed. Refer to the 2275 Test Fixture Kit manual for interface board 21 design and wiring instructions.

4.6.4 Universal Power Supplies (Optional)

The Universal Power Supplies (UPS) provide programmable power to the unit- under-test. Each UPS is programmable over its full-scale voltage range, with a voltage resolution of 11 bits plus sign. Full-scale values available are:

10 V, 4 A	(Requires 1 slot)
20 V, 2 A	(Requires 1 slot)
40 V, 1 A	(Requires 1 slot)
80 V, 0.5 A	(Requires 2 slots)

4.6.5 High-Current Supply (Optional)

The High-Current Supplies (HCS) are program-controlled units with voltage levels and current limits specified by the program. Each HCS consists of a 0-to-7 V, 17-A (0-to-5 V, 35 A also available on 2272 only) Lambda [1] power supply, controlled by optional DC Sources. As an integral part of the unit, 3 relays, energized by relay drivers on the system relay driver board PIO(0), connect or disconnect the HCS outputs from the UUT. See Figure 4-9 for actual supply and control configurations.

The HCS output voltage is the same value as that of the controlling DCS source. Thus, SET DCV DCS(3) V=5 will result in a +5 V HCS(1) output; SET DCV DCS(3) V=-5 results in a -5 V HCS(1) output. The HCS(1) current limit, however, is related to the current limit of the DCI DCS(3) by a scale factor of 100 uA/A. That is, a 100-uA current limit on DCS(3) will result in a 1-A limit on the HCS(1). Also, because of the current-limiting circuit in the HCS units, the DCI DCS(3) current limit must be negative regardless of the output polarity of the HCS(1) (e.g., SET DCI DCS(3) I=-100U will program the HCS(1) output to 1 A). Refer to the 227X Test Programming manual for a description of all program statements controlling DCS.

The HCS output is not bipolar. The actual polarity of the voltage to the UUT depends on the connection of the HI and LO output leads to the test fixture. Refer to the appropriate Test Fixture Kit manual for actual polarity connections.

---

[1] Lambda Electronics Corp., Melville, L.I., N.Y.

The no-load slew rate of the HCS is shown in Figure 4-11. For increasing voltages, the output of the HCS typically slews at a rate of 1 V/ms (no-load). For decreasing voltages, the output decays exponentially through a 3600-uF capacitor and 560-ohm resistor in parallel with the load resistance. Always include appropriate delays in the test program to allow the HCS output to settle.

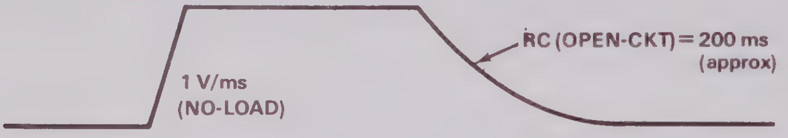


Figure 4-11. HCS Slew Rate (No Load).

4.6.6 Programming Power and Ground Connections

In-circuit testing of analog components requires that no power be applied to the UUT and that the UUT ground not be connected to the test system ground. To perform in-circuit testing of digital ICs, however, both power and ground must be applied to the UUT. Program control of relays via the relay-driver boards (Parallel I/O - PIO) simplifies the connecting and disconnecting of user power supplies and functional ground to the UUT.

Power Connections

UUT power supplies should only be connected to the test fixture through a relay. This allows the UUT to be isolated from the supplies whenever the UUT must be tested without power applied. Program control of the various supplies, therefore, not only consists of designating output voltages and current limits but also includes opening or closing the output relay(s) connecting that supply to the UUT. Table 4-3 summarizes the type of test language statements associated with each supply. Refer to the 227X Test Programming manual for detailed descriptions of the statements.

Table 4-3
POWER-SUPPLY TEST LANGUAGE STATEMENTS

Power Supply	Supply Statements	Relay Statements
Fixed		
2270/71/72	Not Applicable	User Option
2275	SET PIO(0) HRLY(OPEN/CLOSE 12)	SET PIO(0) RLY(OPEN/CLOSE 6)
TPS	SET/MEAS DEVTYPE/CLEAR TPS(n) for slots n = 1, 2, or 3	SET PIO(0) HRLY(OPEN/CLOSE 7)
	SET/MEAS DEVTYPE/CLEAR TPS(n) for slots n = 4, 5, or 6	SET PIO(0) HRLY(OPEN/CLOSE 8)
UPS	SET/TEST/MEAS/CLEAR UPS(n)	SET PIO(0) HRLY(OPEN/CLOSE 8)
HCS	SET/TEST/MEAS/CLEAR DCV DCS(n) SET/TEST/MEAS/CLEAR DCI DCS(n)	SET PIO(0) RLY(OPEN/CLOSE n) SET PIO(0) RLY(OPEN/CLOSE n)
		CLEAR PIO(0)

In general, the standard instrumentation statements SET, TEST, MEAS and CLEAR allow the user to specify the output voltage and current limit, to test or measure the output voltage (or type of supply), and to reset the output to zero. A special MEAS DEVTYPE allows the user to determine the type of supply contained in a particular slot. TPSs may be in slots 1, 2, and 3 or 4, 5, and 6; UPSs can only be in slots 4, 5, and 6. Program control over HCS supplies is via a DCS; thus, the SET/TEST/MEAS/CLEAR statements are addressed to a specific DCS. The SET DCV DCS(3) statement designates the output voltage level of HCS(1), while SET DCI DCS(3) designates the current limit.

NOTE

The user should close all applicable power-supply relays before turning on power and open them after the power-supply outputs are at 0 V.

The relays connected in series with the power-supply outputs are all controlled by relay drivers on the system PIO(0) board. As described in the 227X Test Programming manual, keywords OPEN, CLOSE, etc denote the state to which the specified relay is driven. Relay driver HRLY(7) controls the relay box connected to TPS slots 1, 2, and 3; relay driver HRLY(8) controls the relay box connected to TPS/UPS slots 4, 5, and 6; and relay drivers RLY(15) and RLY(16) control the relays contained in the HCS unit. A CLEAR PIO statement de-energizes all PIO relay drivers, thus opening all relays.

Ground Connections

The 8 parallel ground relays on the relay ground board are controlled by high-voltage relay driver HRLY(3) on the system relay-driver board.

```
SET PIO(0) HRLY(CLOSE 3);
```

closes the relays and connects system ground to all receiver ground pins. Note that all receiver ground pins are internally tied together.

```
SET PIO(0) HRLY(OPEN 3);
```

opens the relays and disconnects system ground from UUT ground.

These 2 test language statements only control the relay ground board and have no effect on other devices on the system relay-driver board PIO(0). Refer to the 227X Test Programming manual for more detail on PIO test language.

Ground pins are tied directly to system ground and are used for coding the test fixture ID and for wiring external circuitry supplied by the user.

Examples

Connecting and disconnecting a UPS and system ground via relay drivers.

```
     SET PIO(0) HRLY(CLOSE 3);      /* Connect system ground */
     SET PIO(0) HRLY(CLOSE 8);      /* Close relay connected to UPS(4) */
     DELAY 50M;
     SET DCV UPS(4) V=5 I=2;            /* Set UPS(4) output voltage */
               |
               |
        digital tests
               |
               |
     SET DCV UPS(4) V=0.0 I=0.2 FAIL();
UPSL4: TEST UPS(4) HI=0.05 FAIL()[BRANCH UPSL4]; /* Wait for Caps */
     CLEAR UPS(4);                  /* Disconnect Power Supply */
     DELAY 50M;
     SET PIO(0) HRLY(OPEN 8);       /* Open relay to UPS(4) */
     SET PIO(0) HRLY(OPEN 3);       /* Disconnect system ground */
```

Meaning: The first SET PIO command closes the ground relays and applies system ground to the UUT. The next SET PIO command activates high-voltage relay driver 8. This relay-driver output then energizes the relay box connected to slots 4, 5, and 6. The 50-ms delay lets the relays settle before UPS power is applied. The SET DCV UPS(4) statement then sets the UPS(4) voltage to 5 volts.

After digital testing is complete, the SET DCV UPS(4) statement resets the power supply to 0 V. The 50-ms delay allows the UUT to discharge before any relay is opened. The subsequent SET PIO commands then de-energize relay HRLY(8), disconnecting the UPS(4) output from the UUT, and open the ground relays, removing system ground from the UUT.

Programming a TPS with relay drivers.

```
SET PIO(0)  HRLY(CLOSE 3);
SET PIO(0)  HRLY(CLOSE 7);
DELAY 50M;
SET TPS(1) VB;
    |
  test statements
    |
CLEAR TPS(1);
DELAY 200M;
SET PIO(0)  HRLY(OPEN 7);
SET PIO(0)  HRLY(OPEN 3);
```

Meaning: Similar to the previous example except that the relay driver for TPS(1) is HRLY 7. The output voltage of a TPS is set by the program resistors. In this case the B level is selected by specifying VB in the SET TPS statement. The TPS can be disconnected with a CLEAR statement since the voltage cannot be set to 0. The voltage is turned off and allows the output to decrease through internal bleed resistors.

Turning on HCS(1).

```
SET DCI DCS(3) I=+10M;        /* Initialize HCS to 0 V and reverse */
SET DCV DCS(3) V=0;           /* current to assure supply output */
                              /* is zero */
DELAY 0.2;                    /* (Example) Delay for HCS zeroing */
SET PIO(0) HRLY(CLOSE 3);     /* Connect system ground */
SET PIO(0) RLY(CLOSE 15);     /* Close HCS output relay */
DELAY 50M;                    /* Wait for relays to close */
SET DCI DCS(3) I=-1M;         /* (Example) Set current limit to 10 A */
SET DCV DCS(3) V=5;           /* (Example) Set output voltage to 5 V */
DELAY 50M;                    /* (Example) Wait for output to settle */
TEST DCV DCS(3)               /* Test for current limit */
 [BRANCH HCSFAIL];
BURST... (Test Program)
```

Meaning: Because the HCS can supply so much power, it must be completely shut off before UUT connection in order to prevent transients: HCS(1) is set to 0 V and negative full-scale current to reverse bias the supply to prevent any possible output. A 200-ms delay allows time for the output to stabilize at zero. The system ground and HCS output relays are closed and 50 ms allowed for the relay contacts to settle. The HCS(1) is then programmed to the desired output current limit and output voltage, and a delay is allowed for output settling. Though 50 ms is a good typical value, the delay can be determined empirically with an oscilloscope monitoring the UUT. The settling time is determined by the difference between the current limit setting and the actual UUT input current, with larger differences allowing quicker settling according to this formula:

$$\text{settling time} = \frac{(3600\ \text{uF} + C) \times \text{set voltage}}{\text{set current} - \text{load current}}$$

where C is the UUT supply filter capacitance in farads.

The current limit should still be set to protect the UUT; the final statement tests for current limiting before testing begins.

Turning off the HCS(1).

```
SET DCI DCS(3) I=+10M;        /* Set HCS output to zero and */
SET DCV DCS(3) V=0;           /* back bias supply to assure zero */
                              /* output */
DELAY 0.2;                    /* Wait for supply to zero */
SET PIO(0) RLY(OPEN 15);      /* Open HCS output relay */
SET PIO(0) HRLY(OPEN 3);      /* Disconnect system ground */
```

Meaning: Setting V=0 and back-biasing the supply turn off HCS(1). A 200-ms delay allows the supply to settle to zero and the UUT filter capacitors to discharge. The HCS(1) output relay and the system ground relay can then be disconnected.

Programming Considerations

1. In digital testing, when power needs to be applied to the board, always connect system ground first and disconnect it last (after the power supplies have been disconnected from the UUT).

2. For all power supplies, but especially for the HCS, only open and close a power-supply relay when the supply output is at 0 V.

3 Be sure to reset all power supplies and open all relays after a test is complete (and prior to starting another test). Otherwise, the relays will open only when the run-time system is being exited.

4. Insert DELAY and TEST statements, as appropriate, to ensure that a power supply has settled to its expected value before relays are opened or closed, components are tested, etc.

5. When using an HCS, where separate statements are needed to specify the current limit and output voltage, typically designate the current limit first, then the output voltage.

6. Some board designs require that voltages be applied and removed in a specific order. The first 5 considerations may be compromised to meet these requirements. The user should always analyze each board type for specific power requirements.

7. Provide for the discharge of circuits at the end of a test, especially where blocking diodes or regulators are present and the power supplies are unable to discharge these nodes. This could prevent damage to the tester.

4.6.7 Relay Drivers

When special tests are being developed, special instrumentation may be connected to the UUT. It is usually not desirable to have the instrumentation connected while other components are being tested. Relays are generally mounted in the test fixture to provide for disconnection. In 2270/71/72 systems, a system-controlled relay-driver board, containing 24 relay drivers, is provided as an option for controlling these relays. The relay-driver board (PIO board) is inserted in PIO slot 1 and its programming name is PIO(1). These relay drivers are typically used to energize relay coils or indicator lamps mounted in the test fixture. In 2275 systems, there are 3 relay drivers on the PIO(0) for user applications. These are RLY(9), RLY(10), and HRLY(5).

The PIO(1) board has 24 relay drivers, each consisting of an open-collector transistor. These drivers have the following capabilities:

Sixteen low-voltage drivers (RLY1L-16L) can switch 100 mA to ground and handle voltages up to 25 V.

Eight high-voltage drivers (HRLY1L-8L) can switch 200 mA to ground and handle voltages up to 60 V.

A typical relay-driver circuit interface is shown in Figure 4-12. The user should include a suppression diode directly across each relay coil to suppress voltage transients produced by the relay coil and to protect the driver

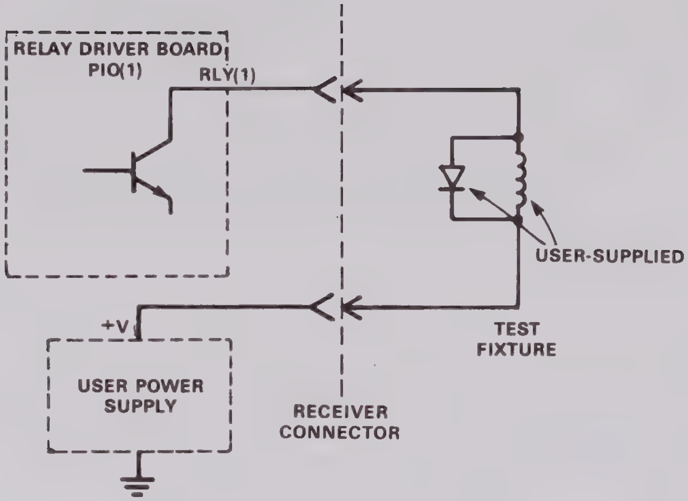

RELAY DRIVER BOARD
PIO(1)
RLY(1)
USER-SUPPLIED
+V
TEST
FIXTURE
USER POWER
SUPPLY
RECEIVER
CONNECTOR

Figure 4-12. Typical Relay Driver Wiring.

transistors when they are turned off. The relay voltage is typically taken from a user power supply. When the driver is used for driving lamp loads, a resistor must be placed in series with the transistor collector and the lamp.

4.6.8 TTL Drivers

Relay-driver board PIO(1) also contains 16 TTL drivers, which can be used to control user-supplied LEDs, non-IEEE bus interfacing or other external circuitry mounted in the test fixture. Each driver can supply up to 1.2 mA at 2.4 V and sink up to 48 mA at 0.4 V. A typical TTL driver interface is shown in Figure 4-13.

The 2275 has 3 TTL drivers available on PIO(0) for user applications. These are TDR(14), TDR(15), and TDR(16).

4.6.9 TTL Sensors

The relay-driver board PIO(0) also contains 24 TTL-level sensors. These circuits (shown in Figure 4-14) can be used to monitor external TTL levels from user circuits, switches or non-IEEE bus instruments. The 24 sensor circuits have the following capabilities:

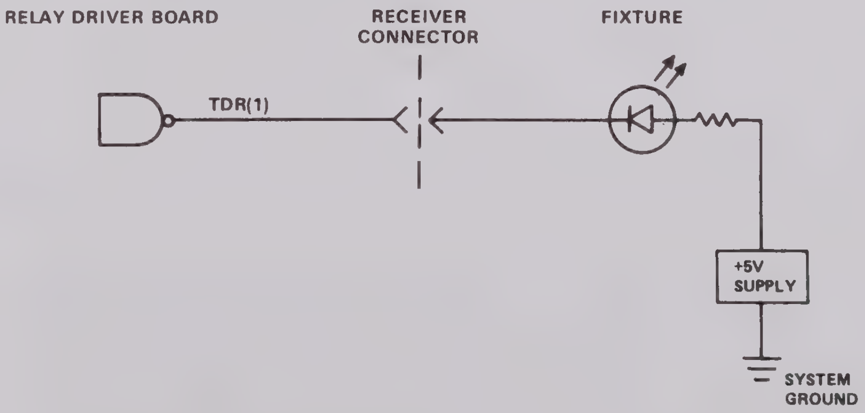

RELAY DRIVER BOARD
RECEIVER
CONNECTOR
FIXTURE
TDR(1)
+5V
SUPPLY
SYSTEM
GROUND

Figure 4-13. Typical TTL Driver Interface.

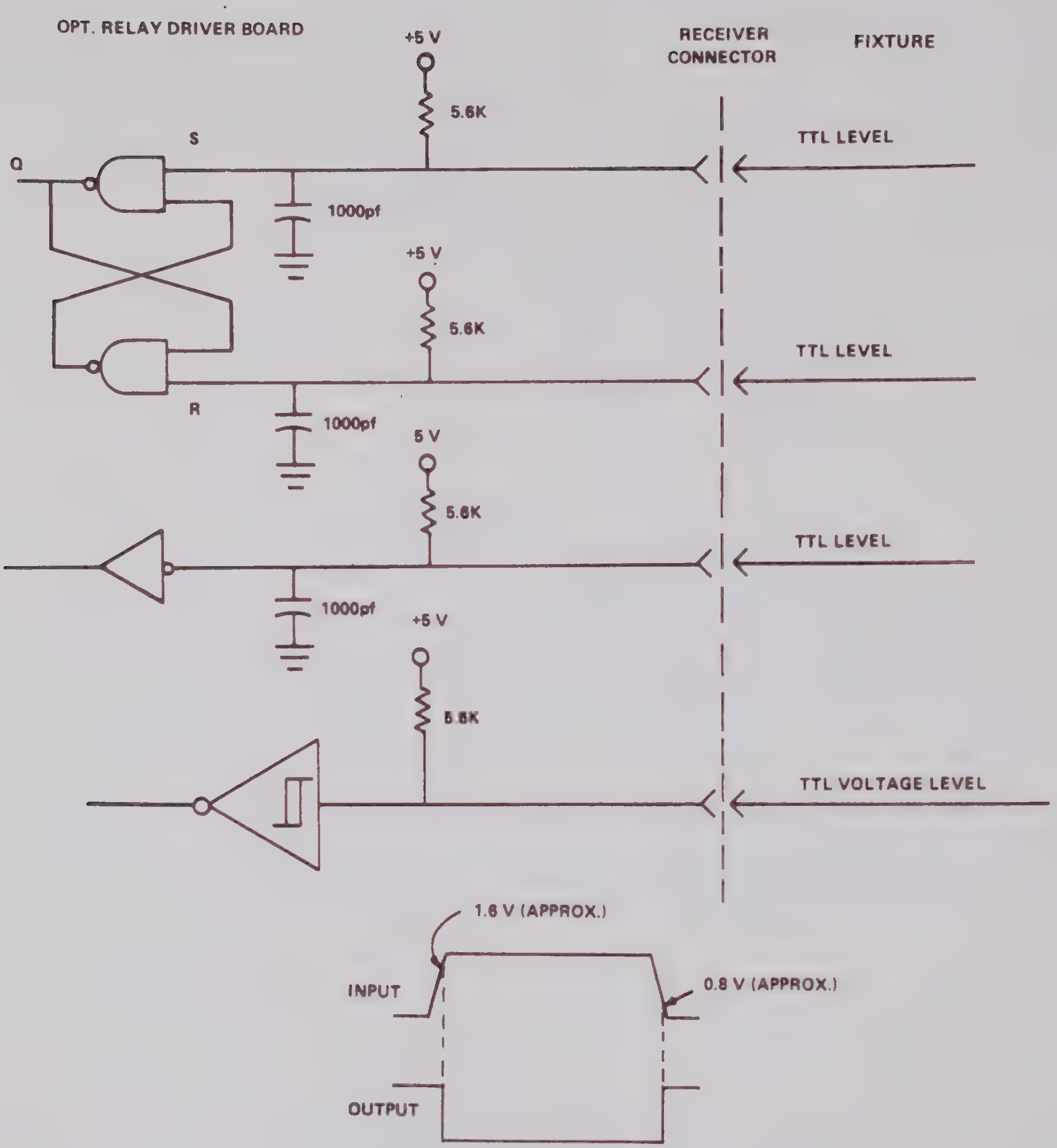

OPT. RELAY DRIVER BOARD
+5 V
RECEIVER CONNECTOR
FIXTURE
5.6K
S
Q
TTL LEVEL
1000pf
+5 V
5.6K
TTL LEVEL
R
1000pf
5 V
5.6K
TTL LEVEL
1000pf
+5 V
5.6K
TTL VOLTAGE LEVEL
1.6 V (APPROX.)
INPUT
0.8 V (APPROX.)
OUTPUT

Figure 4-14. TTL Sensor Wiring.

1. Sixteen sense lines (S1L-S16L) are applied to Schmitt-trigger inverters, each with a typical positive-going threshold of +1.6 V and a negative-going threshold of +0.8 V. Each line drives a single TTL load and is terminated by a 5.6-kilohm resistor to a +5-V supply lead on the relay-driver board.

2. Six pairs of sense lines (SFF1L-6L/RFF1L-6L) are applied to set/reset latches to provide SPDT switch debouncing. Each sense line is terminated by a 5.6-kilohm resistor to a +5-V line and by a 1000-pF capacitor to ground.

3. Two sense lines (SET1L and SET2L) are each applied to the set side of a latch, the reset side of which is disabled. This, in effect, reduces the sensor to a simple TTL inverter. These two lines are also terminated by a 5.6-kilohm resistor to a +5-V line and by a 1000-pF capacitor to ground.

The 2275 has no TTL sensors available for user applications.

4.6.10 Probe

The test system will accommodate a probe that can be used for either analog or digital testing. The probe connects to a dedicated driver/sensor or to external lines that can be switched to the output of the analog instrument multiplexer.

Pressing the ANALOG/DIGITAL switch, on the front panel, to the ANALOG position connects a GenRad probe to output lines EXTA, B, C, D of the Self-Test Module (STM).

A special feature of the GenRad probe is to provide a 2-wire connection very near the probe tip. This makes it possible to make Kelvin connections from any measurement instrument to the actual point of measurement on the UUT, thus eliminating the effects of the probe cable. By comparison, conventional one-wire probes do not allow Kelvin connections to be made any closer than the probe switch panel and then only when a special shorting plug is inserted in the probe connector.

For user convenience and to accommodate other probes, a separate probe test point and BNC connector are provided which, when the ANALOG/DIGITAL switch is in the ANALOG position, connect directly to external channels A and C (and indirectly to channels B and D via the GenRad probe tip). Appropriate SET MUX statements can then be used to connect any system analog instrument to the MUX channels and thus to the probe. Refer to the 227X Test Programming manual for the test language statements required to control the STM and the MUX.

Releasing the ANALOG/DIGITAL switch to the DIGITAL position connects the GenRad probe, test point and BNC connector to a dedicated driver/sensor (via the fixture wiring). The user can be directed to use the probe to check continuity between the pins of a digital IC and the system driver sensors.

4.6.11 Strobes

Several internal pulses (strobes) from the digital driver/sensor system are available at the front panel for easy access to the user. These strobes may be used for synchronizing external equipment to system timing during debugging of programs or for special testing requirements. All workstations have the 4 standard strobes: TEST STEP 1, TEST STEP N, DRIVER, and SENSOR; see Figure 4-15. Workstations with the SpeedPlus option also have a panel with 6 additional high-speed strobes: RUN, TEST STEP, TRIGGER, SENSOR, DRIVER, and CLOCK. In all cases the strobe wiring consists of parallel wires bought out to a BNC connector and a test point.

The timing associated with all test steps in a digital burst is controlled by the ITT and IST parameters specified by the user in a BURST statement. IST (Interstrobe Time) defines the time interval between driving the input pins and sensing the output pins in a test step. ITT (Inter-Test Time) defines the time interval between consecutive test steps.

The use of the standard strobes and SpeedPlus strobes is described in Tables 4-4 and 4-5, respectively.

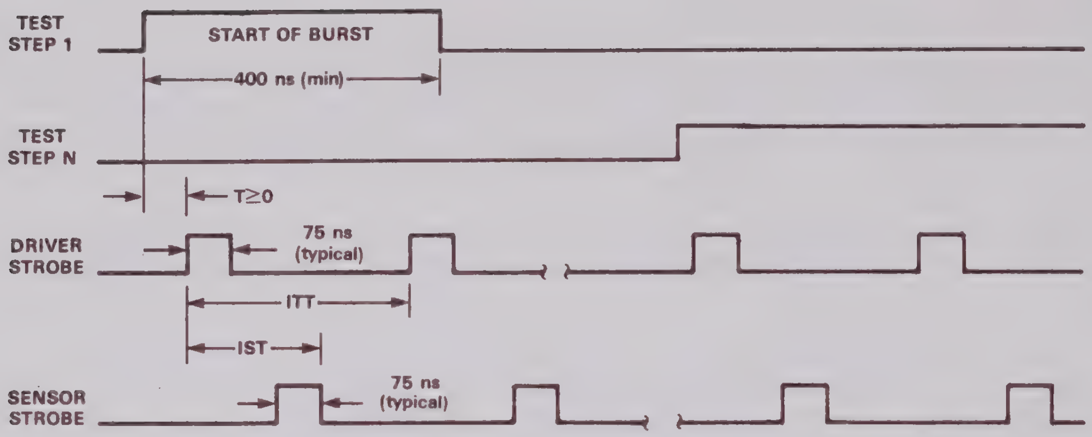


Figure 4-15. Sync Pulse Timing.

Table 4-4
STANDARD STROBE DESCRIPTIONS

Strobe	Description
TEST STEP 1	This strobe occurs at the beginning of each burst and is synchronized with the driver strobe timing. The TEST STEP 1 pulse is developed even if no driver or sensor strobe pulse is generated at that step.
TEST STEP N	This strobe is generated when a TEST LGC or SET LGC statement appears to change the state of the selected "step N" test pin at the desired test step.
DRIVER/SENSOR	The DRIVER strobe is an internal pulse used to parallel-load pin specification data into all driver/sensors. The SENSOR Strobe is used to latch the output state of all sensors after a test. These 2 strobe pulses occur at each TEST LGC or SET LGC statement, except during a set LGC statement specifying the driver/sensor terminating resistors (i.e., $ PU, $ PD, $ PN, or $ PB pin specifications). A SET LGC statement directed to these pull-up and pull-down resistors generates neither strobe but still causes an ITT delay before the next test step. Refer to the 227X Test Programming manual for details on digital testing and digital test statements.

Table 4-5
SpeedPlus STROBE DESCRIPTIONS

Strobe	Description
RUN	This strobe occurs at the beginning of a high speed portion of a burst. It is synchronized with the driver strobe timing and is generated whether or not a driver or sensor strobe exists in that step.
TEST STEP	This strobe occurs as a result of a TEST LGC or SET LGC at a step in the high-speed burst.

Table 4-5 (Continued)

Strobe	Description
DRIVER/SENSOR	Same as standard DRIVER/SENSOR strobes except occurs during high-speed bursts or during slave mode in standard bursts.
TRIGGER	This strobe occurs when trigger sample is taken and trigger is enabled.
CLOCK	This strobe occurs at the clock sync pulse.

4.6.12 Optional DC Source

There may be situations where additional dc voltage and current sources could improve a test. An optional dc voltage source and an optional dc current source are available which are connected to the 4 instrumentation channels through an optional instrumentation multiplexer. Figure 4-16 illustrates the

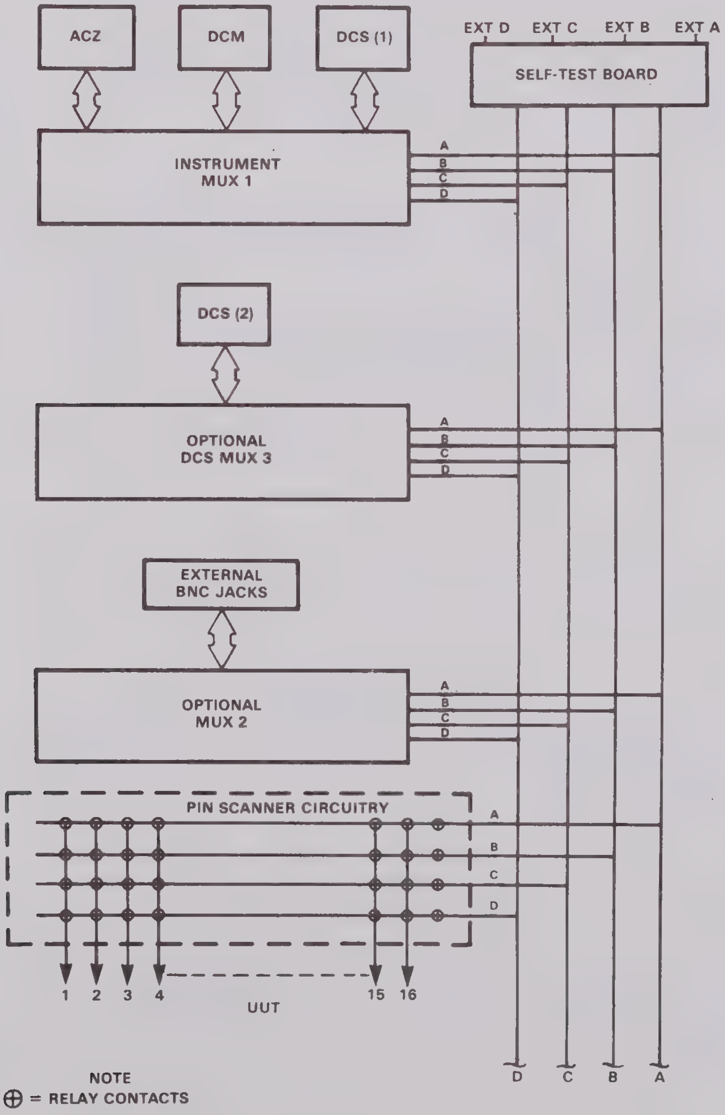

ACZ
DCM
DCS (1)
EXT D
EXT C
EXT B
EXT A
SELF-TEST BOARD
INSTRUMENT MUX 1
A
B
C
D
DCS (2)
OPTIONAL DCS MUX 3
A
B
C
D
EXTERNAL BNC JACKS
OPTIONAL MUX 2
A
B
C
D
PIN SCANNER CIRCUITRY
A
B
C
D
1
2
3
4
UUT
15
16
D
C
B
A
NOTE
⊕ = RELAY CONTACTS

Figure 4-16. Instrumentation Channels.

configuration of all standard and optional instrumentation that can be connected to the measurement channels.

Examples of where an additional source may be helpful are:

Using 2 voltage sources, instead of a voltage and a current source, for certain transistor tests.
Providing separate voltage sources on the inputs of an op-amp or differential amplifier.
Providing power for a low-current circuit.

Refer to the Test Language manual for information on programming the DCS.

4.6.13 IEEE Instrumentation Options

Functional testing with external instrumentation is only limited by the user's needs and creativity and by the fact that there are 4 instrumentation channels through which external signals can be applied to the pc board for any single test. See Figure 4-16. Refer to the 227X Test Programming manual for information on programming IEEE instruments.

With this option, the user may connect any external IEEE 488 compatible, programmable instrument to the instrumentation bus by providing a BNC cable (customer-supplied) from each device-addressable output to one of the 9 BNC jacks (J1-J9) mounted on the IEEE panel in the rear of the support bay.

As shown in Figure 4-17, each instrument must also be connected to the interface bus through the use of IEEE bus cables that connect from the 24-pin connector on the instrument to the IEEE Board. The cable that plugs into the interface board is supplied with the option. This cable can be ordered in 2- or 4-meter lengths. Inter-device cables are customer-supplied but are available from GenRad in 1-, 2-, and 4-meter lengths. However, the maximum recommended bus length is 2 meters times the number of devices, but not more than 20 meters. The device cable connectors are double-ended so that linear or star configurations can be made, as shown in Figure 4-18.

NOTE
Bus Interrupt and Priority to the IEEE bus must be maintained via Bus Grant cards or other system modules, as shown in the Diagrams manuals.

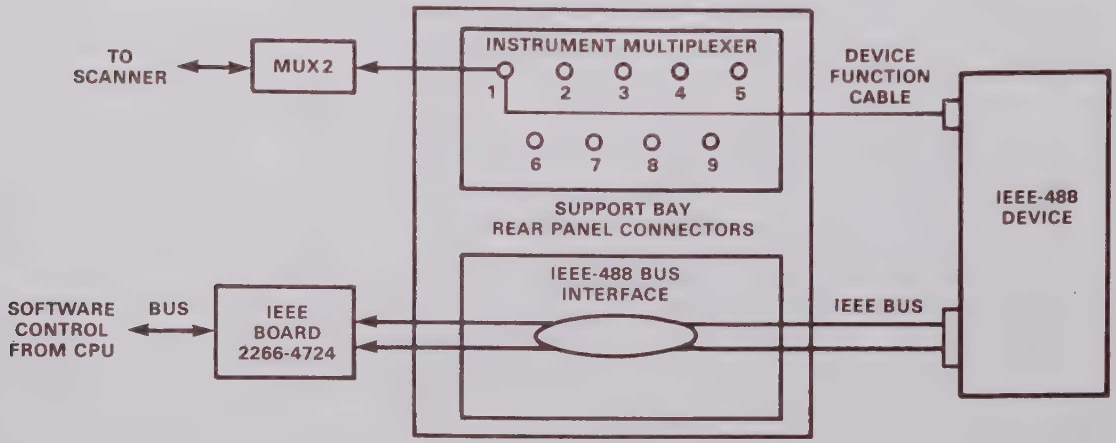


Figure 4-17. IEEE-488 Device Connections to the System.

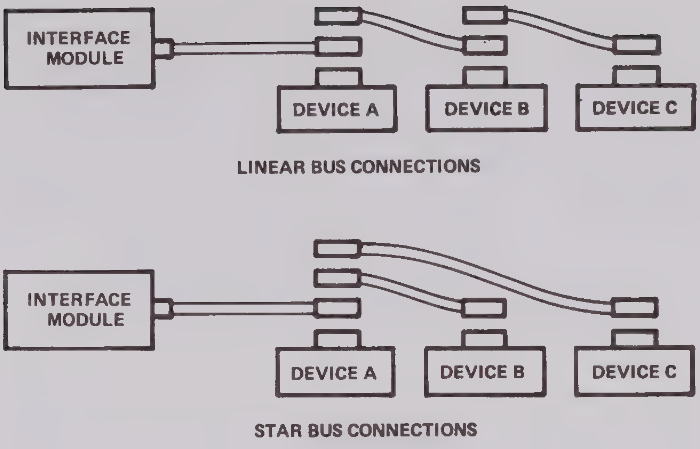

INTERFACE MODULE
DEVICE A
DEVICE B
DEVICE C
LINEAR BUS CONNECTIONS
INTERFACE MODULE
DEVICE A
DEVICE B
DEVICE C
STAR BUS CONNECTIONS

Figure 4-18. IEEE-488 Bus Connector.

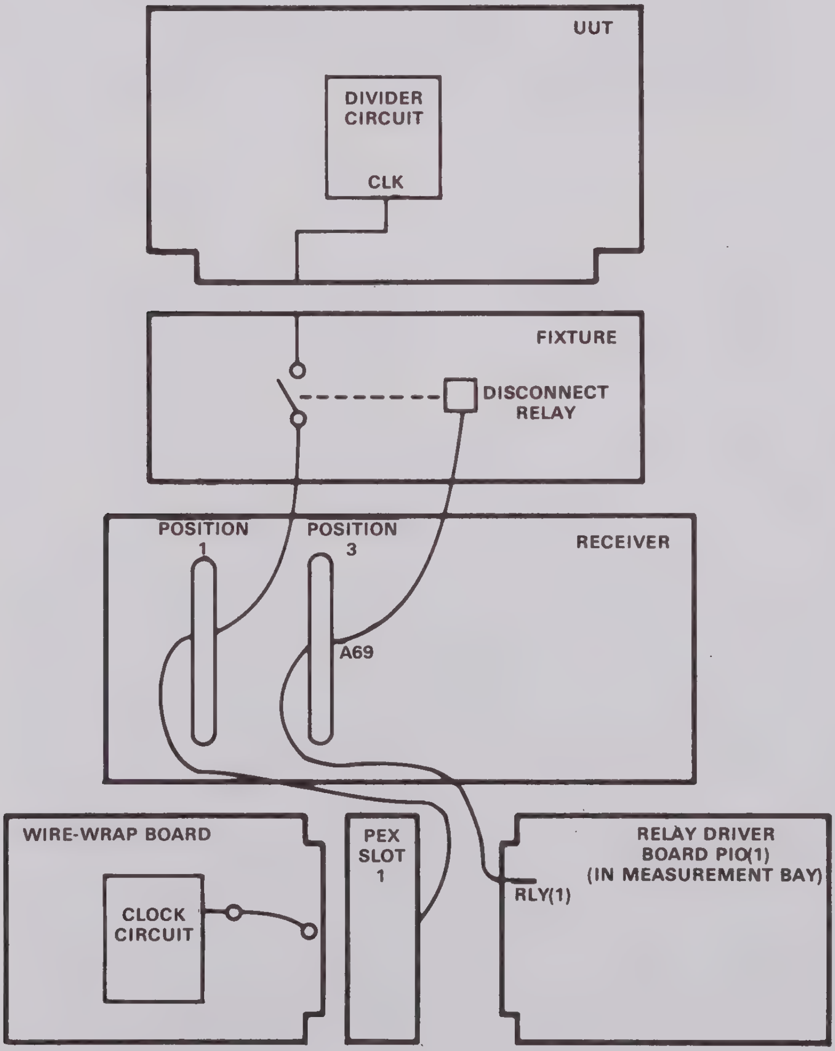

UUT
DIVIDER CIRCUIT
CLK
FIXTURE
DISCONNECT RELAY
POSITION 1
POSITION 3
RECEIVER
A69
WIRE-WRAP BOARD
CLOCK CIRCUIT
PEX SLOT 1
RELAY DRIVER BOARD PIO(1) (IN MEASUREMENT BAY)
RLY(1)

Figure 4-19. User-Provided Clock on Wire-Wrap Board.

4.6.14 Optional Instrumentation Board

The 2272 provides the ability to add instrumentation, of the user's design, to the system. This is done via an optional wire-wrap board (P/N 2272-1558) that plugs into position 1 and/or 2 of the pin expander bay. Direct connections exist between the wire-wrap board connector and position 1 on the receiver. These lines can be wired directly to a test fixture nail or to a disconnect relay added to the fixture. The optional PIO(1) can then be used to control the relay. See Figure 4-19 for a typical example of a user-added instrument, an external clock, being used to test a counter on the UUT.

4.7 SpeedPlus OPTIONAL FEATURES

There are optional features available for 2271 and 2272 test workstations which provide programmable clock drivers, clock synchronization, and event triggers. Library routines, for devices that should be tested at operating speed, contain clock drive statements. The user can modify these statements with .ATO commands. ATO commands are also used when an on-board clock exists. The ATO clock programming choices available for dealing with on-board clocks are: disable the on-board clock and use the system clock, use the on-board clock and synchronize to it, do not do a fast test, and do not do any test. ATO commands are described in Section 5.

4.8 FIXTURE DESIGN

4.8.1 General

Fixture design is coordinated with the test programming steps in accordance with the strategy selected for the fixture design. If the strategy is to provide fixturing in the minimum time, perform the steps for drilling the nail holes in parallel with the programming effort. In the case of a 2270 system, the wiring can also be performed in parallel with programming. An alternate strategy minimizes rewiring of the test fixture if ECO changes are made to the pc board. This alternate approach provides for putting nails on the output end of all applicable runs of the pc board. To do this, the Nail Assignment Report (.NAR) needs to be provided before the fixture drilling template is designed, because the .NAR lists output pins and other component leads that are preferred nail locations. The fixture designer should refer to the Test Fixture Kit Assembly manual for a detailed discussion of the reasons and procedures of the various strategies of fixture design.

Fixture converters are available for fixtures designed for using (1) a 2270 fixture on a 2271 system, (2) a 2270 on a 2272, (3) a 2271 on a 2272, (4) a 2275 on a 2271, and (5) a 2275 on a 2272.

Once the fixture nails are assembled in the fixture, the wiring of the fixture can begin. The power-supply wiring and special wiring are determined by system configuration and test program requirements. Information for this comes from an understanding of power connections described in the Test Fixture Kit Assembly manual and from specifications provided by the test program writer. The remainder of the fixture wiring information is in the form of a Fixture Wire-wrap Report, also called the Nail Fixture Report (.NFR). This report is generated by the nail assignment task that the test program writer will run in order to complete the .TPG. Refer to Section 8 for a complete discussion.

4.8.2 Fixture Drilling Requirements

The nail socket holes must be drilled in the top plate of the fixture so that a nail can be mounted in the fixture for each node on the etch. Fixture manufacturers have different requirements for determining the location of each nail hole. An accurate 1:1 Mylar pc layout drawing with nail locations drawn on it may be sufficient. This could be taken one step further by making a drilling drawing which would be easier to use. A numerical controlled (NC) drilling tape could also be made if NC equipment is to be used. Some manufacturers may prefer a bare board with the node locations marked on it. Check with the fixture vendor for details.

4.8.3 Known-Good Boards

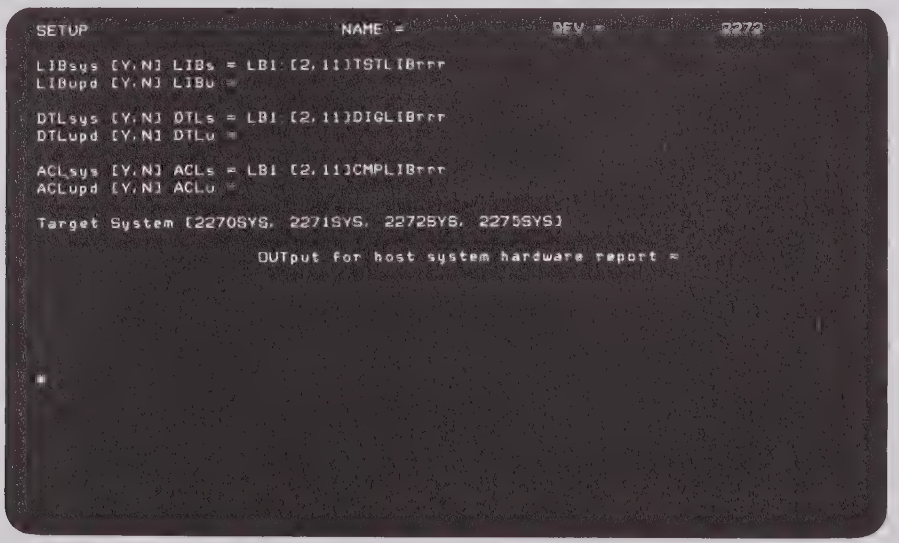
Known-good boards are used for debugging the test program and test fixture. Several are required so that faults can be inserted to check out fault detection. Some boards may be destroyed by the debugging process.

4.9 SETUP UTILITY

The Setup utility supplies the names of the system and update library files used by ATG and, when run, produces a report on the hardware configuration of the host system. See Figure 4-20.

GenRad test systems have separate libraries of test information for different component types. The Analog Test Library contains tests for analog components such as resistors, capacitors, inductors, etc. The Digital Test Library contains tests and data for many digital ICs. The Analog Component Library provides tests for complex and hybrid ICs which combine analog and digital circuitry. Complete information about the test libraries is found in the Test Library Programming manual.

Each library type can have 3 actual library files: the system library, the update library, and the user library. Two of these files, the system and the update libraries, are provided by GenRad with each system. The user library file



```
SETUP                         NAME =                    DEV =              2272

LIBsys [Y,N] LIBs = LB1:[2,11]TSTLIBrrr
LIBupd [Y,N] LIBu =

DTLsys [Y,N] DTLs = LB1:[2,11]DIGLIBrrr
DTLupd [Y,N] DTLu =

ACLsys [Y,N] ACLs = LB1:[2,11]CMPLIBrrr
ACLupd [Y,N] ACLu =

Target System [2270SYS, 2271SYS, 2272SYS, 2275SYS]

                     OUTput for host system hardware report =
```

Figure 4-20. Setup Options Page.

is created by the test programmer to contain custom-component tests not included in the system or update libraries. Options on the Setup options page (Figure 4-20) provide a way to block ATG from using the system and update libraries. The default setting for all the library options is Y, indicating the libraries do exist. Setting a library option to N instructs the system not to look for that specific library file and removes the associated file specification from the page.

LIBsys [Y,N] LIBs =

Selects whether or not a system library file from the set of Analog Test Libraries (.LIB) is to be used by ATG in writing the test program. The 3 characters, designated rrr in Figure 4-20, indicate the release number.

[Y] The default for this option, Y, indicates a system library exists and is to be used by ATG. When Y is enabled, the associated file spec option, "LIBs =," appears.

[N] Selecting N indicates either the system library file does not exist or that it is not to be used by ATG when writing the test program. When N is enabled, the associated file spec option does not appear on the page.

LIBupd [Y,N] LIBu =

Selects whether or not an update library file from the set of Analog Test Libraries (.LIB) is to be used by ATG in writing the test program.

[Y] The default for this option, Y, indicates an update library exists and is to be used by ATG. When Y is enabled, the associated file spec option, "LIBu =," appears on the line.

[N] Selecting N indicates either the update library file does not exist or that it is not to be used by ATG when writing the test program. When N is enabled, the associated file spec option does not appear on the page.

DTLsys [Y,N] DTLs =

Selects whether or not a system library file from the set of Digital Test Libraries (.DTL) is to be used by ATG in writing the test program.

[Y] The default for this option, Y, indicates a system library exists and is to be used by ATG. When Y is enabled, the associated file spec option, "DTLs =," appears.

[N] Selecting N indicates either the system library file does not exist or that it is not to be used by ATG when writing the test program. When N is enabled, the associated file spec option does not appear on the page.

DTLupd [Y,N] DTLu =

Selects whether or not an update library file from the set of Digital Test Libraries (.DTL) is to be used by ATG in writing the test program.

[Y] The default for this option, Y, indicates an update library exists and is to be used by ATG. When Y is enabled, the associated file spec option, "DTLu =," appears on the line.

[N] Selecting N indicates either the update library file does not exist or that it is not to be used by ATG when writing the test program. When N is enabled, the associated file spec option does not appear on the page.

ACLsys [Y,N] ACLs =

Selects whether or not a system library file from the set of Analog Component Test Libraries (.ACL) is to be used by ATG in writing the test program.

[Y] The default for this option, Y, indicates a system library exists and is to be used by ATG. When Y is enabled, the associated file spec option, "ACLs =," appears.

[N] Selecting N indicates either the system library file does not exist or that it is not to be used by ATG when writing the test program. When N is enabled, the associated file spec option does not appear on the page.

ACLupd [Y,N] ACLu =

Selects whether or not an update library file from the set of Analog Component Test Libraries (.ACL) is to be used by ATG in writing the test program.

[Y] The default for this option, Y, indicates an update library exists and is to be used by ATG. When Y is enabled, the associated file spec option, "ACLu =," appears on the line.

[N] Selecting N indicates either the update library file does not exist or that it is not to be used by ATG when writing the test program. When N is enabled, the associated file spec option does not appear on the page.

Target System [2270SYS, 2271SYS, 2272SYS, 2275SYS]

Selects the model of 227X series of board test workstations for which the test set is being prepared.

OUTput for host system hardware report =

Specifies the file or the output device on which the hardware configuration report will appear. Setting OUT = TI: displays the report on the video screen; setting OUT = LP: (or LP1: for systems with the optional line printer) prints the configuration report. An example of this report is shown in Figure 4-21.

```
Hardware Configuration of 2272:
--------------------------------

Board Type     Boards
----------     ------
PEX       :     1 - 12   (max nail = 1536)
D/S       :     1 - 12
Scanner   :     1 - 12
DCS       :     1 , 3
DCM       :     1
ACZ       :     1
MUX       :     1 - 3
PIO       :     0 , 1
STM       :     1
TPS       :     1 - 3
UPS       :     4         (20 volts)
UPS       :     5         (20 volts)
UPS       :     6         (40 volts)
BCR       :     1
HSE:DIG   :     1 - 8    (max nail = 4256)
  GR IEEE interface present
```

Figure 4-21. Configuration Report.

This report is interpreted to show hardware exists as follows:

PEX boards 1 through 12 (128 test points per board)
D/S boards 1 through 12
Scanner boards 1 through 12
DCS 1 (standard dc source)
DCS 3 (used to control HCS 1)
DCM 1 (standard dc measurement unit)
ACZ 1 (standard ac impedance measurement unit)
MUX 1 (standard instrument MUX)
MUX 2 (for optional DCS 2)
MUX 3 (for IEEE instrument)
PIO 0 (standard relay-drive board)
PIO 1 (optional relay-drive board)
STM 1 (self-test module)
TPS in slots 1 through 3
UPS in slots 4 amd 5 (20-volt universal supply)
UPS in slot 6 (40-volt universal supply)
BCR (bar-code reader is present)
HSE:DIG (SpeedPlus, high-speed driver/sensor 1 through 8)
GR IEEE interface board present

4.10 RESULTS OF TEST SET ANALYSIS

Once the pc board test analysis is done, assignments can be given to test programmers to generate any special routines. These assignments should include a system configuration report and circuit schematic and parts data.

If it has been decided that new system hardware options are required, purchase requisitions should be prepared.

Finally, when all special programming is complete, a report should be created indicating any .ATO or library files that were created so that the appropriate names can be entered on the ATG options page. This is also important for keeping up to date the documentation on the contents of the Test Set. The report should also indicate how to treat the special components in the .CKT file. This includes type, value, and flagspec information.

Automatic Test Options Programming – Section 5

5.1 INTRODUCTION

In the "test plan" phase of test set preparation it may have been decided that treatment of special circuit conditions and/or the setting of standard power statements in the ATG test program would be inserted in the test program by providing an automatic test options (.ATO) file. In order to have made this decision, the user would have to know what can be done with an .ATO file and how it works. This section describes the "what" and the "how" of .ATO files.

Basically an .ATO file replaces certain statements that would otherwise be inserted into the test program by ATG and/or it includes additional statements. These actions are done at specific locations in the program that are definable by .ATO commands. At the time that ATG-analog and ATG-digital are run, the ATG task automatically looks for an .ATO file with the same name as the test program name and which has an extension .ATO or .ATX. ATG uses those statements indicated in the .ATO file instead of and/or in addition to those provided in the various libraries. The process is illustrated in Figure 5-1.

An .ATO file must be created by the user by invoking a system editor (described in Section 11). The name of the file must be the same as the name of the test program and must have the extension .ATO or .ATX.

The difference between .ATO and .ATX files has to do with multiplexed and non-multiplexed test points on the various 227X systems. The 2270 has non-multiplexed test points and all others have multiplexed test points. Input files for multiplex systems must have temporary nail assignments. These temporary nail assignments exist in temporary files which use the .--X file extension. These temporary files become updated with real test points and with real files extensions after ATG by running the nail assignment procedure. Thus, when creating a file for a 2270 use the extension .ATO. When creating a file for a 2271, 2272, or 2275, use the extension .ATX. Throughout this section the term .ATO is used to mean .ATO and/or .ATX. The reader should understand which is applicable to the target system for which the test set is being generated. Figure 5-2 illustrates the relationship of .ATO files to the ATG and Nail Assignment processes.

An automatic test options file may contain any number of commands (options) that apply to a single test program. The last command in an .ATO file must be the keyword END.

The primary command types, REPLACE and INCLUDE, are used for both digital options and analog options. The format and usage varies, so their descriptions are given in para 5.2 for digital procedures and in para 5.3 for analog procedures. Paragraph 5.4 is an illustration of the .ATO file used for the TRAINGBD test program.

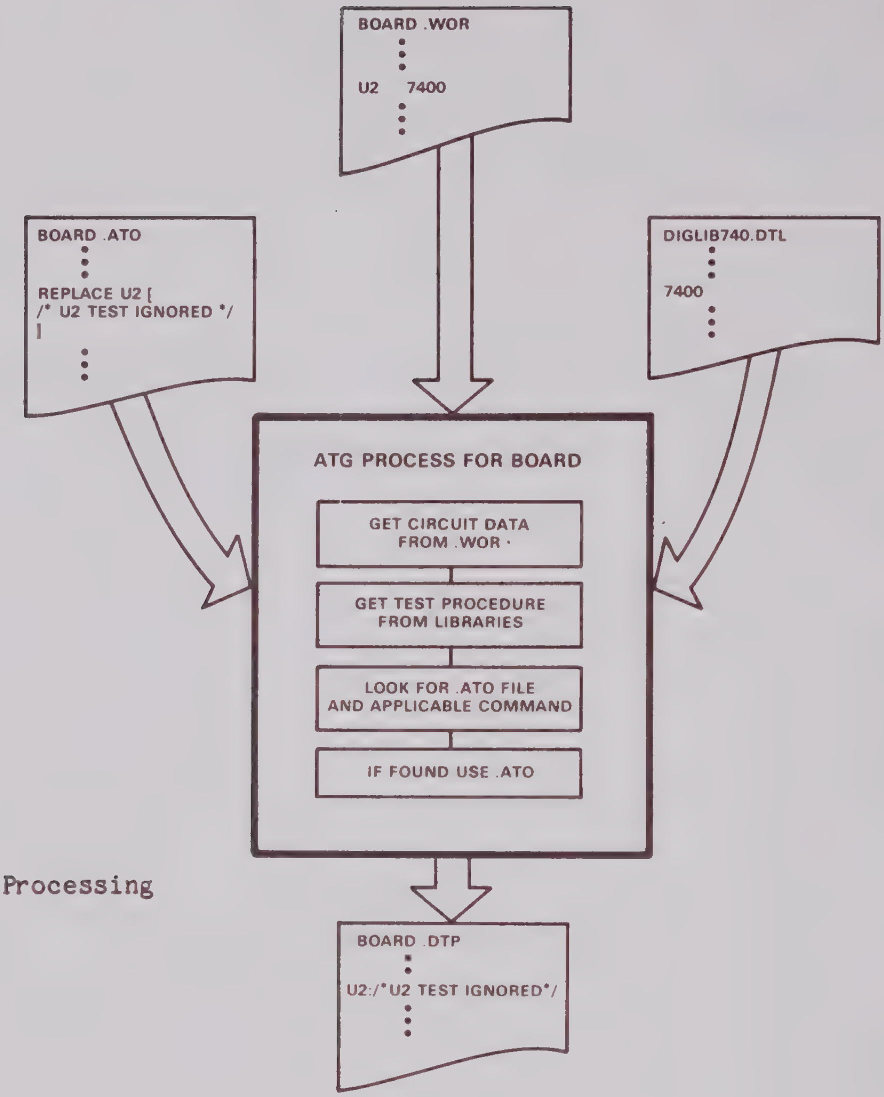

BOARD .WOR
U2 7400
BOARD .ATO
REPLACE U2 [
/* U2 TEST IGNORED */
]
DIGLIB740.DTL
7400
ATG PROCESS FOR BOARD
GET CIRCUIT DATA FROM .WOR
GET TEST PROCEDURE FROM LIBRARIES
LOOK FOR .ATO FILE AND APPLICABLE COMMAND
IF FOUND USE .ATO
BOARD .DTP
U2:/*U2 TEST IGNORED*/

Figure 5-1. ATG Processing of an .ATO File.

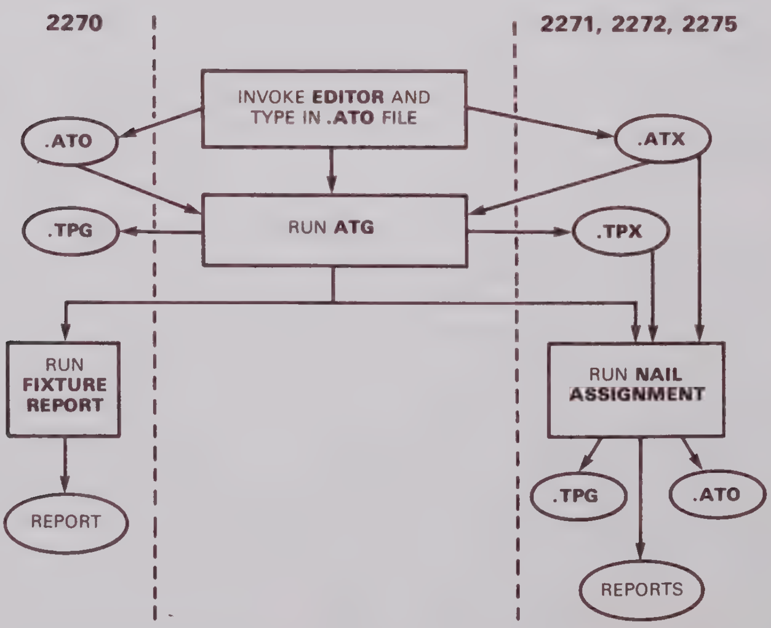

2270
2271, 2272, 2275
INVOKE EDITOR AND TYPE IN .ATO FILE
.ATO
.ATX
RUN ATG
.TPG
.TPX
RUN FIXTURE REPORT
REPORT
RUN NAIL ASSIGNMENT
.TPG
.ATO
REPORTS

Figure 5-2. Relationship of .ATO to ATG and NAIL ASSIGNMENT for Multiplexed and Non-Multiplexed Test Point Systems.

5.2 WRITING DIGITAL .ATO COMMANDS

A digital .ATO command consists of a command keyword followed by a location identifier followed by the subject matter of the command. The subject matter must be enclosed in brackets ([]).

```
command-keyword location-identifier [subject matter]
```

The digital command-keywords and acceptable location-identifiers, and subject-matter, associated with each are:

Command-Keyword	Location-Identifiers	Subject-Matter
REPLACE	ALLBURST BUSINIT BUSEND BUSTEST DIGINIT DIGEND FORCES 1 or more program labels	any test statement(s)
INCLUDE	same as REPLACE	any test statement(s)
FLAG	1 or more program labels	NOFAST NOTEST
CLOCK	1 or more program labels	SET CLOCK AT statement FREQ statement ENABLE statements DISABLE statements

5.2.1 REPLACE Command

The digital REPLACE command directs ATG to insert statements other than those that ATG would normally insert into the test program. The program location for the replacement is indicated by the location identifier. The location identifier is either one or more program labels such as U2: or one of the following identifier keywords:

ALLBURST
BUSINIT
BUSEND
BUSTEST
DIGINIT
DIGEND
FORCES

The statements inserted are written between brackets ([]) and may be any number of complete test language statements or any comments. Digital replace command examples:

```
REPLACE BUSINIT [
     ASSIGN LGC VIHA=4.5 VILA=0.0 VOHA=2.2 VOLA=1.4 LVLA=(1-16);
]
```

Meaning: The standard logic statement that ATG inserts before a bus test section is replaced with the logic statement that includes only the bussed test points (1-16).

```
REPLACE FORCES [
     HFORCE NAME='U3';
        IC(27,30,36-42) IL(27, 36-42);
        IH(30);
]
```

Meaning: The FORCE statements that ATG would normally insert as a result of FORCE statements in the digital library, are replaced with these FORCE statements in all BUSTESTS.

Definitions of Location-identifier Keywords

Location-identifier keywords used in digital REPLACE and INCLUDE commands are:

ALLBURST

The ALLBURST identifier refers to the beginning of every BURST or BUSTEST in a digital test program and every independent section within a BURST. An independent section of a burst is delimited by the condition where all the inputs are disconnected and all outputs are ignored. If the command is INCLUDE, then the test language statements will be included just prior to the first statement of the BURST (or just following the statement BURST;) and just prior to the first statement of each independent section of the burst (these sections are defined in the library procedure). If the command is REPLACE, every independent part of every BURST is replaced with the test language statements included in the command. REPLACE ALLBURST is not a practical command.

BUSEND

The BUSEND identifier refers to a location between the completion of bus tests and the start of tests for bused ICs. The INCLUDE command would cause the test language statements to be included at this point. The REPLACE command would insert the .ATO statements in place of those normally inserted by DTG.

BUSINIT

The BUSINIT identifier refers to a location just prior to the start of bus tests. The INCLUDE command would cause the test language statements to be included at this point. The REPLACE command would insert the .ATO statements in place of those normally inserted by DTG.

BUSTEST

The BUSTEST identifier refers to the beginning of a BUSTEST sequence. The use of REPLACE BUSTEST [/* Suppress Bustests */] will turn off all bus test processing and insert the commented text in its place.

DIGEND

The DIGEND identifier refers to the group of standard test statements (such as power down and relay disconnect) that are normally included by DTG at the end of a digital test program. The INCLUDE command would add the test language statements after those normally included by DTG. The REPLACE command would insert the .ATO statements in place of those normally inserted by DTG.

DIGINIT

The DIGINIT identifier refers to the group of standard test statements (such as relay connect and power set) that are normally included by DTG at the beginning of a digital test program. The INCLUDE command would add the test language statements after those normally included by DTG. The REPLACE command would insert the .ATO statements in place of those normally inserted by DTG.

FORCES

The FORCES identifier refers to the beginning of the force section of a bustest. The statement INCLUDE FORCES [/* comment */] will include the bracketed comment in front of the force text of all bustests. The statement REPLACE FORCES [/* comment */] will turn off force statements in the bustests, reducing them to simple go/nogo tests on the buses. The bracketed comment will be inserted in front of the disabled force text.

Figure 5-3 shows the locations and extent of the various REPLACE and INCLUDE commands in a typical digital test program.

5.2.2 INCLUDE Command

The digital INCLUDE command directs ATG to insert statements in addition to those ATG would normally insert into the test program. The program location for these added statements is indicated by the location identifier. The location identifiers are identical to those used in REPLACE commands. The statements inserted are written between brackets ([]) and may be any number of complete test language statements or any comments.

Digital INCLUDE command examples:

```
INCLUDE ALLBURST [$IC(73) IL(73);       /* DISABLE CLOCK */ ]
```

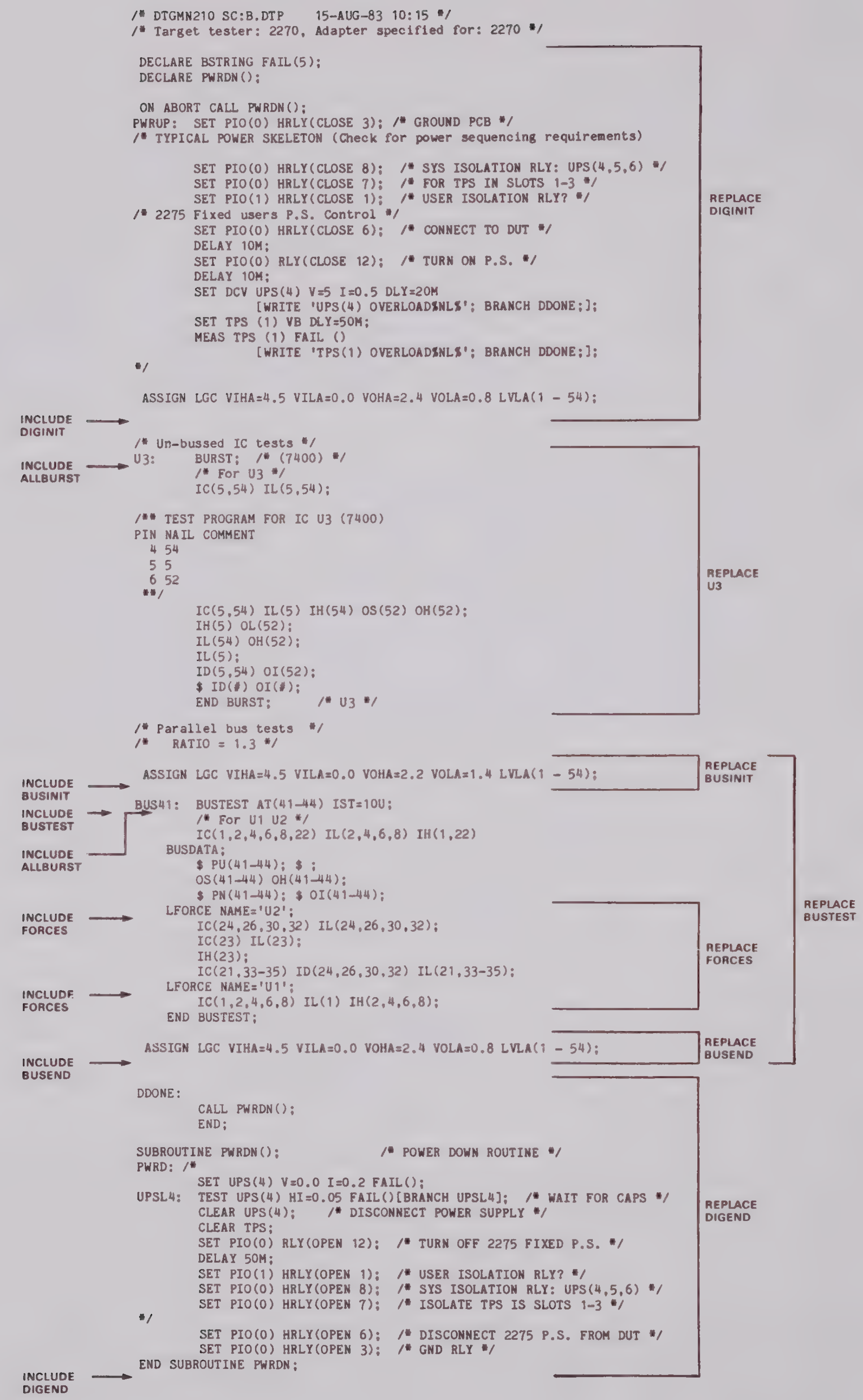
Meaning: An on-board clock needs to be disabled for all digital testing. Holding nail 73 low during each burst accomplishes the disable. The disabling command and comment, are inserted at the beginning of every burst.

5.2.3 FLAG Command

There are two types of digital FLAG commands. One type directs ATG to select the normal speed test from the digital library rather than the fast test. This assumes that the library contains both a fast and a normal test for the device. If no normal test is provided in the library, the effect of the FLAG would be to not do any test for the device. The test system must have the Speed Plus option for this command to be appropriate. The other type directs ATG to refrain from inserting any test into the program for the specified device. The location identifier is one or more program labels. The subject matter is one of the two keywords that designate which type of FLAG command, which are:

NOFAST and NOTEST.

The location identifier must be enclosed in parentheses and the type keyword must be enclosed in brackets.



```
/* DTGMN210 SC:B.DTP    15-AUG-83 10:15 */
/* Target tester: 2270, Adapter specified for: 2270 */

 DECLARE BSTRING FAIL(5);
 DECLARE PWRDN();

 ON ABORT CALL PWRDN();
PWRUP:  SET PIO(0) HRLY(CLOSE 3); /* GROUND PCB */
/* TYPICAL POWER SKELETON (Check for power sequencing requirements)

        SET PIO(0) HRLY(CLOSE 8);  /* SYS ISOLATION RLY: UPS(4,5,6) */
        SET PIO(0) HRLY(CLOSE 7);  /* FOR TPS IN SLOTS 1-3 */
        SET PIO(1) HRLY(CLOSE 1);  /* USER ISOLATION RLY? */
/* 2275 Fixed users P.S. Control */
        SET PIO(0) HRLY(CLOSE 6);  /* CONNECT TO DUT */
        DELAY 10M;
        SET PIO(0) RLY(CLOSE 12);  /* TURN ON P.S. */
        DELAY 10M;
        SET DCV UPS(4) V=5 I=0.5 DLY=20M
                [WRITE 'UPS(4) OVERLOAD$NL$'; BRANCH DDONE;];
        SET TPS (1) VB DLY=50M;
        MEAS TPS (1) FAIL ()
                [WRITE 'TPS(1) OVERLOAD$NL$'; BRANCH DDONE;];
*/

 ASSIGN LGC VIHA=4.5 VILA=0.0 VOHA=2.4 VOLA=0.8 LVLA(1 - 54);

/* Un-bussed IC tests */
U3:     BURST;  /* (7400) */
        /* For U3 */
        IC(5,54) IL(5,54);

/** TEST PROGRAM FOR IC U3 (7400)
PIN NAIL COMMENT
  4 54
  5 5
  6 52
 **/
        IC(5,54) IL(5) IH(54) OS(52) OH(52);
        IH(5) OL(52);
        IL(54) OH(52);
        IL(5);
        ID(5,54) OI(52);
        $ ID(#) OI(#);
        END BURST;      /* U3 */

/* Parallel bus tests  */
/*   RATIO = 1.3 */

 ASSIGN LGC VIHA=4.5 VILA=0.0 VOHA=2.2 VOLA=1.4 LVLA(1 - 54);

BUS41:  BUSTEST AT(41-44) IST=10U;
        /* For U1 U2 */
        IC(1,2,4,6,8,22) IL(2,4,6,8) IH(1,22)
    BUSDATA;
        $ PU(41-44); $ ;
        OS(41-44) OH(41-44);
        $ PN(41-44); $ OI(41-44);
    LFORCE NAME='U2';
        IC(24,26,30,32) IL(24,26,30,32);
        IC(23) IL(23);
        IH(23);
        IC(21,33-35) ID(24,26,30,32) IL(21,33-35);
    LFORCE NAME='U1';
        IC(1,2,4,6,8) IL(1) IH(2,4,6,8);
    END BUSTEST;

 ASSIGN LGC VIHA=4.5 VILA=0.0 VOHA=2.4 VOLA=0.8 LVLA(1 - 54);

DDONE:
        CALL PWRDN();
        END;

SUBROUTINE PWRDN();              /* POWER DOWN ROUTINE */
PWRD: /*
        SET UPS(4) V=0.0 I=0.2 FAIL();
UPSL4:  TEST UPS(4) HI=0.05 FAIL()[BRANCH UPSL4];  /* WAIT FOR CAPS */
        CLEAR UPS(4);    /* DISCONNECT POWER SUPPLY */
        CLEAR TPS;
        SET PIO(0) RLY(OPEN 12);  /* TURN OFF 2275 FIXED P.S. */
        DELAY 50M;
        SET PIO(1) HRLY(OPEN 1);  /* USER ISOLATION RLY? */
        SET PIO(0) HRLY(OPEN 8);  /* SYS ISOLATION RLY: UPS(4,5,6) */
        SET PIO(0) HRLY(OPEN 7);  /* ISOLATE TPS IS SLOTS 1-3 */
*/
        SET PIO(0) HRLY(OPEN 6);  /* DISCONNECT 2275 P.S. FROM DUT */
        SET PIO(0) HRLY(OPEN 3);  /* GND RLY */
 END SUBROUTINE PWRDN;
```

REPLACE DIGINIT
INCLUDE DIGINIT
INCLUDE ALLBURST
REPLACE U3
INCLUDE BUSINIT
REPLACE BUSINIT
INCLUDE BUSTEST
INCLUDE ALLBURST
REPLACE BUSTEST
INCLUDE FORCES
REPLACE FORCES
INCLUDE FORCES
REPLACE BUSEND
INCLUDE BUSEND
REPLACE DIGEND
INCLUDE DIGEND

Figure 5-3. Test Program Locations Affected by Digital .ATO Commands.

Digital FLAG command example:

```
FLAG (U1) [NOFAST     /* USE NORMAL TEST */]
```

Meaning: ATG is to insert, in the DTP, the normal rather than the fast test procedure for device U1.

5.2.4 CLOCK Command

The primary reason for providing CLOCK commands in an .ATO file is to provide a means for dealing with on-board clocks and to provide clock drive signals that simulate the DUT circuit. CLOCK commands can provide for synchronizing tester clocks and driver/sensor activity with the on-board clock. A means of enabling and disabling an on-board clock can also be specified.

The CLOCK command location identifier consists of one or more digital component names (program labels). The subject matter is a clock rate or a clock statement consistent with the definition of clock statements given in the 227X Test Programming manual. This clock statement would be used in place of the clock statement that exists in the library routine for each device listed in the location identifier.

When only a clock rate is provided, ATG generates a default SET CLOCK statement using a rate equal to (or the closest lower rate obtainable) that specified. The default statement is consistent with the defaults described in the 227X Test Programming manual.

In the case of an on-board clock, the CLOCK command consists of a location description, and any combination of one or more of the following 3 items: clock statement, enable section, and disable section. The clock statement would provide information for synchronizing driver/sensor activity to the on-board clock. The enable section would describe how to enable the on-board clock, if it is not always running. The disable section describes how to disable the on-board clock for testing components that do not need the clock. Enable statements are inserted in all routines specified in the location description and the disable statements are inserted in all other routines.

ENABle DISABLE sections are identical to the library ENABLE and DISABLE sections described in the 227X Test Library Programming manual.

Examples

1. Assume the pc board has a processor, U4, and there is a library routine for that type of processor. The library routine has a clock statement that sets the clock rate at 2 Megahertz. The processor is used with a 3-Megahertz clock rate on the pc board and it is desirable to test at operating speed. The default values for all other clock characteristics are acceptable. The .ATO command to make the change is

```
CLOCK (U4) [FREQ = 3MEG]
```

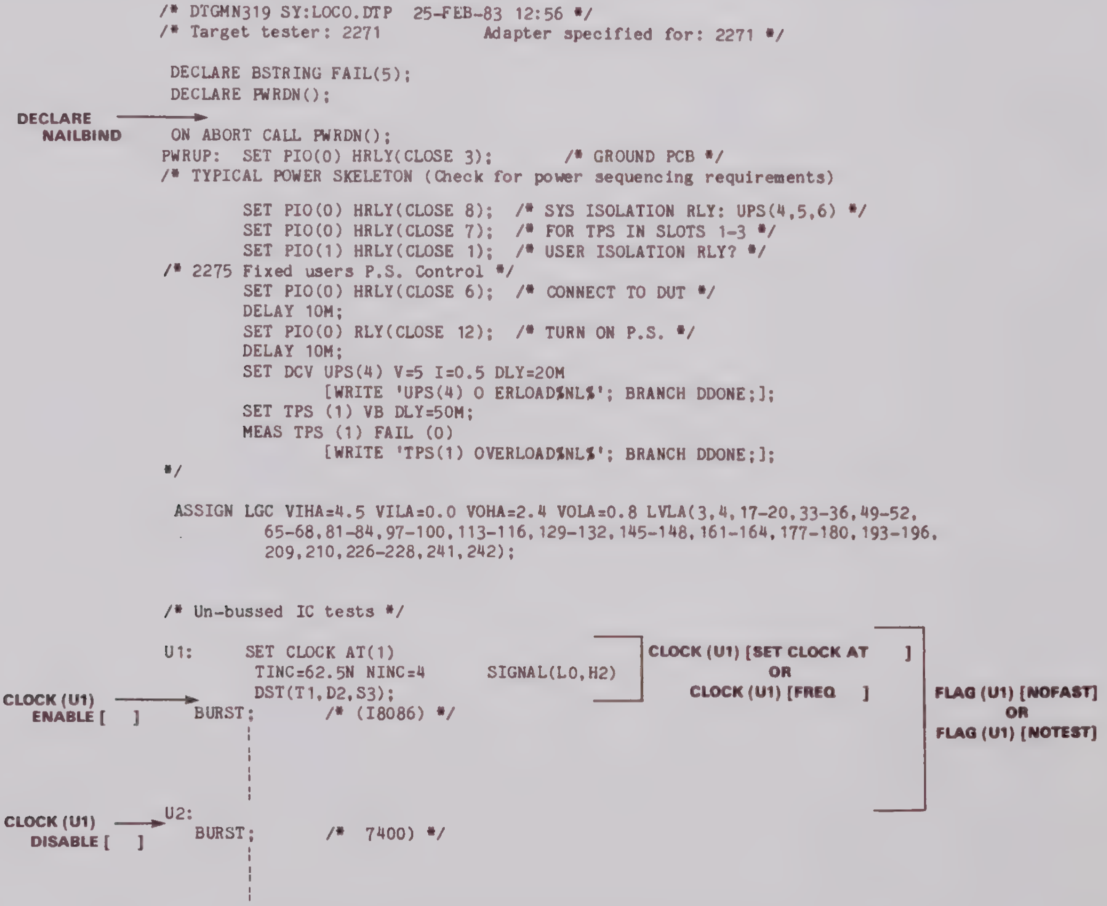
2. Assume the pc board has an on-board clock which is used when testing ICs U1, U2 and U3 and must be disabled while testing all other ICs. The .ATO command to insert a clock disable instruction at all ICs except U1, U2, and U3 is

```
CLOCK (U1, U2, U3)

    DISABLE [.............]
```

3. Assume the pc board has an on-board clock. It is desired to synchronize to the on-board clock so that driver/sensor timing and voltage can be controlled for the various test nodes. If the synchronizing nail is 6, the clock period is 500 nanoseconds with 2 increments per period, and the sync signal is low for increment 0 and high for increment 1, then the .ATO command to replace the clock statement that is in the library for the digital devices would be

```
CLOCK (U1) [SET CLOCK AT (CSYNC=6) NINC=2 PERIOD=500NS CSYNC(LO, HI);]
```

Figure 5-4 shows the program locations affected by CLOCK and FLAG commands in a typical digital test program.



```
/* DTGMN319 SY:LOCO.DTP  25-FEB-83 12:56 */
/* Target tester: 2271          Adapter specified for: 2271 */

 DECLARE BSTRING FAIL(5);
 DECLARE PWRDN();

 ON ABORT CALL PWRDN();
PWRUP:  SET PIO(0) HRLY(CLOSE 3);       /* GROUND PCB */
/* TYPICAL POWER SKELETON (Check for power sequencing requirements)

        SET PIO(0) HRLY(CLOSE 8);  /* SYS ISOLATION RLY: UPS(4,5,6) */
        SET PIO(0) HRLY(CLOSE 7);  /* FOR TPS IN SLOTS 1-3 */
        SET PIO(1) HRLY(CLOSE 1);  /* USER ISOLATION RLY? */
/* 2275 Fixed users P.S. Control */
        SET PIO(0) HRLY(CLOSE 6);  /* CONNECT TO DUT */
        DELAY 10M;
        SET PIO(0) RLY(CLOSE 12);  /* TURN ON P.S. */
        DELAY 10M;
        SET DCV UPS(4) V=5 I=0.5 DLY=20M
                [WRITE 'UPS(4) O ERLOAD%NL%'; BRANCH DDONE;];
        SET TPS (1) VB DLY=50M;
        MEAS TPS (1) FAIL (0)
                [WRITE 'TPS(1) OVERLOAD%NL%'; BRANCH DDONE;];
*/

 ASSIGN LGC VIHA=4.5 VILA=0.0 VOHA=2.4 VOLA=0.8 LVLA(3,4,17-20,33-36,49-52,
          65-68,81-84,97-100,113-116,129-132,145-148,161-164,177-180,193-196,
          209,210,226-228,241,242);


/* Un-bussed IC tests */

U1:     SET CLOCK AT(1)
         TINC=62.5N NINC=4      SIGNAL(LO,H2)
        DST(T1,D2,S3);
   BURST;       /* (I8086) */

U2:
   BURST;       /*  7400) */
```

Figure 5-4. Locations Affected by SpeedPlus .ATO Commands.

5.3 WRITING ANALOG .ATO COMMANDS

An analog .ATO command consists of a command keyword followed by a location identifier followed by a stream number, followed by the subject matter of the command. The stream number must be enclosed in parentheses and the subject matter must be enclosed in brackets ([]).

```
command-keyword location-identifier (stream number) [subject matter]
```

The command keyword can be REPLACE or INCLUDE.

5.3.1 REPLACE Command

The REPLACE command directs analog ATG to use the statements listed in the command instead of the library routine, that would otherwise have been used for a particular device. ATG groups analog tests such that all tests of the same type are listed together. The group is identified by a number called a stream number. This stream number as well as the location identifier must be included in the command. The location identifier must be a program label. The stream number must be in accordance with those shown in the stream list given in Section 7, Table 7-4. Stream number 1 cannot be subject to a replace.

A component may be tested with routines in more than one stream. Separate commands would be required to replace each test. If only one command for one of the streams is provided, no test is inserted in the other stream.

Examples:

Assume that capacitor C54 could cause problems if charged up during test. Therefore, it is desired to have ATG avoid inserting a test for it. The .ATO file entry to do this would include a comment about the entry and a REPLACE command that includes only a comment that the component cannot be tested. The following entry causes ATG not to generate a test for C54 and to insert the comment in stream 42, which is the stream where all guarded capacitance tests are written.

```
/* AVOID CHARGING C54 */
REPLACE C54 (42) [
/* C54 CANNOT BE TESTED */
]
```

Assume that an RF choke exists in the circuit and it is not likely that ATG will find a good test for it. A continuity test can be inserted in place of both the inductance test and the resistance test with the following command.

```
/* SIMPLIFY TESTING OF RF CHOKE */
REPLACE L12 (27)[
/* L12 INDUCTANCE NOT MEASURED */
/* TESTING UNGUARDED RESISTANCE ONLY */
MEAS R AT (53:55)
  DMSG='2.5'
  CMSG='L12'
  LO=0.1
  HI=5.0;
]
```

5.3.2 INCLUDE Command

The INCLUDE command directs ATG to add the statements listed in the command to test program stream. The addition can only be at the beginning of the stream or at the ending of the stream as indicated by the location identifier BEFORE or AFTER.

Example:

> Assume that the test personnel would like to know when capacitance testing is being performed. A message could be displayed when capacitance testing begins and another message displayed when capacitance testing ends by writing the following into the .ATO file.

```
/* MARK BEGINNING OF STREAM 40 */
INCLUDE BEFORE (40) [
WRITE '%NL%NOW TESTING CAPACITORS%NL%';
]
/* MARK END OF STREAM 46 */
INCLUDE AFTER (46) [
WRITE '%NL%FINISHED TESTING CAPACITORS%NL%';
]
```

5.4 TRAINGBD.ATX

The user training pc board has several device types that are not in any system library. Assume the test set preparation plan specified that the ATO file would provide tests for the devices listed in Table 5-2.

Table 5-2
DEVICES TESTED BY TRAINGBD.ATX FILE

Name	Type	Description
R19	R	Low value resistance requiring Kelvin measurement
DS1	CR	Diode that cannot be tested by type CR library routine
U5	DA	D/A converter that needs a special test
U9 & U10		Trigger circuits that need special tests

The board also contains an oscillator which is not to be tested and requires a disable so that the rest of the board can be tested. This disable statement will be provided in the ATO file.

Special force instructions are needed for use in BUSBUST testing. These FORCES will be provided in the ATO file.

An example of an ATO file used for the training board (TRAINGBD.ATX) is presented here. The test procedures (.ATP and .DTP) are shown in Section 7 so that they can be inspected to see how the ATO procedures were actually placed.

TRAINGBD.ATX

Comments

```
/*  TRAINGBD.ATX FILE CREATED BY GENRAD ON 26 OCT 83 FOR
    THE CUSTOMER TRAINING BOARD.

    THIS .ATX POVIDES TESTS FOR THE FOLLOWING DEVICES:
        1 OHM RESISTOR R19
        LED1
        DAC U5 NE5118
        OP-AMP U6 LM1458
        NON-SPEEDPLUS TEST FOR CPU U2 Z80
*/
```

The heading identifies the board name, file, creator, date, and devices tested by the ATO procedures.

```
INCLUDE BEFORE (1)  [

        DECLARE PWRDN();
                                 DECLARE FAILCRLED ();
                                 DECLARE LOOP;
DECLARE GROUP DACDATA(206,205,244,243,242,241,240,239);
DECLARE FLOAT DACOUT1;              DECLARE FLOAT DACOUT3;
DECLARE FLOAT DACOUT2;              DECLARE FLOAT DACOUT4;
DECLARE FLOAT VREFOUT;

       ]
```

This ATO command inserts declare statements, in the beginning of the declare stream, to declare variable names and subroutine names used by ATO procedures.

```
REPLACE R19 (39)   [

/* =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
         */
/* KELVIN MEASUREMENT PROCEEDURE FOR R19 */
/* =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

         */

/* IN THIS PROCEEDURE, THE MULTIPLEXER IS CONFIGURED WITH THE DC VOLTAGE
 MEASUREMENT UNIT ACROSS THE DEVICE UNDER TEST WITH ITS REFERENCE LINE
 LEFT FLOATING. THIS IS DONE BY PUTTING (P+) ON CHC AND (P--) ON CHD
 THEN THE DC VOLTAGE SOURCE (VHI) IS CONNECTED TO CHA. THE DC CURRENT
 MEASUREMENT UNIT (IIN) IS THEN CONNECTED TO CHB. THE REFERENCE LINES
 OF THE DCS (REF5 AND REF6) ARE THEN CONNECTED TO THE REFERENCE LINE OF
 THE DC CURRENT MEASUREMENT UNIT (REF8) AND CONNECTED TO INSTRUMENTATION
 GROUND WITH REF-11. THIS TEST PROCEEDURE REQUIRES TWO TESTER NAILS ON
 EACH SIDE OF THE DUT  ONE PAIR OF NAILS ARE CONNECTED TO THE DC VOLTAGE
 MEASUREMENT UNIT AND THE OTHER PAIR OF NAILS ARE USED TO CARRY THE TEST
 CURRENT. THE CURRENT FLOW THROUGH THE INSTRUMENTATION AND THE DUT IS
 MEASURED AS IS THE VOLTAGE DROP ACCROSS THE DUT. USING THE FORMULA

 RDUT = VDUT / I
 THE ACTUAL VALUE OF THE DUT IS MEASURED ACCURATLY BY BYPASSING THE
 NORMALLY ENCOUNTERED SYSTEM RESIDUALS. */

        /*KELVIN MEAS FOR 1 OHM */
        USE FSUB=FAILSUB,
        R19:    SET MUX AT(CHA=6. CHB=8: CHC=4: CHD=5: REF=5,6,8,11);
                SET SCAN AT(48:46 47:45);
                SET DCV DCS V=400M I=128M DLY=30M ,
                MEAS DCV INTO VVAL1 DLY=30M,
                MEAS DCI INTO IVAL1 DLY=30M,
                LET VVAL2=(VVAL1/IVAL1);
                TEST ARITH VVAL2 HI=1.33 LO=0 9 CMSG='R19' DMSG='1 OHM


          /* END OF KELVIN TEST */
 ]
```

This ATO command replaces the normal resistance test with the Kelvin measurement procedure needed to test the 1 ohm resistor R19. The test is put in stream 39 which is a stream number not used by ATG but is assigned for this test by the ATO command.

The procedure is well documented to explain the test. The test requires 2 nails on each node that the resistor is connected to. The .CKT specifies the 2 nails on each node.

```
 REPLACE DS1 (51)     [

 SETLED:          /* THIS IS THE SET UP FOR TESTING L.E.D.
         CHA ---VOLTAGE MEAS. & CURRENT SOURCE
         CHB ---VOLTAGE SOURCE
         CHC ---GND
         CHD ---GND

         SET SCAN AT (ANODE:CA:CA:CA)
                                                                 */
         SET MUX AT (4,7:5,10:10:REF=4,7,11);



     USE FSUB=FAILCRLED;

 DS1:    /* DS1       */
         SET SCAN AT (80:45:45:45);
         SET DCI DCS I=14.5M;
         TEST DCI DCS FAIL(5);
         IF FAIL(5) THEN BRANCH LED5E;
         MEAS DCV DCM INTO VVAL1  HI=5 LO=4.3 MAX=615.385M
         CMSG='DS1'   DMSG='FWD VOLTAGE'
            PMSG='  LED P/N 6565 1222 ' ;
 LED5E:  SET DCI DCS I=0.0 OFF;

 ]
```

The L.E.D., DS1, cannot be tested with the CR test provided in the library. Therefore, this ATO procedure will replace the CR test that ATG normally puts in stream 51.

TRAINGBD.ATX (continued)

Comments

```
INCLUDE AFTER (87)                [

        WRITE 'NOW TURNING UP POWER%NL%',

        ON ABORT CALL PWRDN();
PWRUP:  SET PIO(0) HRLY(CLOSE 3);        /* GROUND PCB */
        SET PIO(0) HRLY(CLOSE 6); /* CONNECT FIXED P.S. TO DUT */
        DELAY 10M;
        SET PIO(0) RLY(CLOSE 12);        /* TURN ON FIXED P.S. OUTPUTS */
        DELAY 10M,
        SET MUX AT (4·5);                /* CONNECT DCVM TO MUX */

        SET SCAN AT (15 :16 );
        TEST DCV DCM VAL=5.0 TOL=5 DLY=5M [
                WRITE 'INCORRECT VOLTAGE SETTING OF +5V %NL%',
                BRANCH DDONE; ];

        SET SCAN AT (18:16);
        TEST DCV DCM VAL=-15 TOL=10 DLY=5M [
                WRITE 'INCORRECT VOLTAGE SETTING OF -15V %NL%';
                BRANCH DDONE; ];

        SET SCAN AT (17:16);
        TEST DCV DCM VAL=15 TOL=10 DLY=5M [
                WRITE 'INCORRECT VOLTAGE SETTING OF +15V %NL%';
                BRANCH DDONE; ];


                WRITE 'NOW TESTING OPAMP (U6)%NL%',

SETOPAMP:/*THE MUX CONNECTIONS FOR THE OPAMP TEST ARE

 CHA = POSITIVE SIDE OF THE VOLTAGE SOURCE
 CHB = MINUS SIDE OF THE VOLTAGE SOURCE
 CHC = POSITIVE SIDE OF THE VOLTMETER
 CHD = NEGATIVE SIDE OF THE VOLTMETER
 REF 5 AND 6 CONNECT THE NEGATIVE SIDE OF THE VOLTAGE SOURCE TO CHB
*/

        SET MUX AT (6:10:4:5:5,6);
        USE FSUB=FAILCOMP;

/*THE VOLTAGE SOURCE IS CONNECTED TO THE INVERTING AND NONINVERTING
 INPUTS OF THE OPAMP. THE VOLTMETER IS CONNECTED BETWEEN THE OUT-
 PUT OF THE OPAMP AND THE NEGATIVE SUPPLY. THIS IS A "POWER ON"
 TEST. THEREFORE, THE POWER SUPPLY STATEMENTS MUST BE EXECUTED
 BEFORE THE TESTS ARE PERFORMED. A SMALL POSITIVE DIFFERENTIAL
 VOLTAGE IS APPLIED TO THE INPUTS. THIS WILL SEND THE OUTPUT OF
 THE OPAMP TO THE NEGATIVE RAIL VOLTAGE. THE VOLTMETER SHOULD READ
 APPROXIMATELY 0 VOLTS. NOW, A SMALL NEGATIVE VOLTAGE IS APPLIED
 TO THE INPUTS AND THE OUTPUT WILL GO TO THE POSITIVE RAIL VOLTAGE
 THE VOLTMETER SHOULD NOW MEASURE THE TOTAL VOLTAGE SWING OF THE
 OPAMP. FOR EXAMPLE, WITH +15 VOLT AND -15 VOLT SUPPLIES, THE
 METER WILL MEASURE 30 VOLTS. IF THE OPAMP IS OPERATING WITH A
 SINGLE POWER SUPPLY, ONLY ONE RAIL MEASUREMENT WILL BE MADE.*/

]
```

Normally the power turn-on procedure is inserted by ATG just before the digital test program (.DTP). In the case of the training board, power is needed for testing the op-amp, U6. This command inserts a power turn-on procedure at the end of stream 87 which is the last analog stream with which ATG is normally concerned.

```
REPLACE U6 (102)  [

/* MODIFIED OP-AMP TEST FOR DEVICE U6 WITH BRANCHING STATEMENT THAT
 ABORTS THE DAC (U5) TEST IF ITS OUTPUT DEVICE (U6) IS BAD */

U6: USE PMSG='DUAL LM1458';
         SET SCAN AT (117:118:181:18);
         SET DCV DCS V=100M I=20M DLY=0;
         TEST DCV DCS DMSG='UNABLE TO APPLY POSITIVE INPUT VOLTAGE' FAIL(8)
                [BRANCH U6E; ];
         MEAS DCV DCM HI=3 LO=-3 MAX=1
                DMSG='NEGATIVE RAIL' FAIL(8) [BRANCH U6E; ];
         SET DCV DCS V=-100M I=20M DLY=0;
         TEST DCV DCS DMSG='UNABLE TO APPLY NEGATIVE INPUT VOLTAGE' FAIL(8)
                [BRANCH U6E; ];
         MEAS DCV DCM HI=33 LO=20 DLY=5M DMSG='POSITIVE RAIL' FAIL(8)
         [BRANCH U6E; ];
U6E:    SET DCV DCS V=0 I=20M;
         IF FAIL(8) THEN [WRITE ID=MESFILE'
        OUTPUT DEVICE FOR DAC U5 FAILED, ABORTING FURTHER TEST OF THIS%NL%';
        WRITE ID=MESFILE'
        DEVICE AND ALSO ABORTING TEST OF DAC U5%NL%';
         BRANCH U5_13; ];
```

This procedure is the op-amp test which uses the previously inserted power procedure. It replaces any op-amp test found in the libraries. The reason for using a modified op-amp test is so that the DAC test can be by-passed if the op-amp test fails. The DAC test depends on the op-amp operating properly. There is a BRANCH U5__13; statement in the procedure which causes test program execution to jump to the end of the DAC test if the op-amp test fails. The normal stream for op-amp tests is 102.

TRAINGBD.ATX (continued)

Comments

```
U6_1:
          SET SCAN AT (77:181:21:18);
          SET DCV DCS V=100M I=20M DLY=0;
          TEST DCV DCS DMSG='UNABLE TO APPLY POSITIVE INPUT VOLTAGE'
                 [BRANCH U6E; ];
          MEAS DCV DCM HI=3 LO=-3 MAX=1
                 DMSG='NEGATIVE RAIL' [];
          SET DCV DCS V=-100M I=20M DLY=0;
          TEST DCV DCS DMSG='UNABLE TO APPLY NEGATIVE INPUT VOLTAGE'
                 [BRANCH U6E; ];
          MEAS DCV DCM HI=33 LO=20 MAX=30 DMSG='POSITIVE RAIL'
                 [];
U6_1E:    SET DCV DCS V=0 I=20M;
          CLEAR SCAN;     CLEAR MUX;

/* END OF MODIFIED OP-AMP TEST */

]

 INCLUDE BEFORE (110)                    [

 /* INSERTS THE DAC TEST AFTER THE OP-AMP TESTS */

 /* THE FOLLOWING TEST IS FOR THE NE5118 DIGITAL TO ANALOG CONVERTER, THE
  TEST PROCEEDURE FIRST CHECKS THE DAC'S INTERNAL VOLTAGE REFERENCE TO
  INSURE THAT IT IS WITHIN ACCEPTABLE LIMITS. IF THE REFERENCE VOLTAGE
  TEST FAILS, FAILURES IN THE REST OF THE DAC'S TEST PROCEEDURES MAY BE
  ATTRIBUTED TO FAULTY REFERENCE REGULATION. THE NEXT PROCEEDURE CHECKS
  FOR THE ABILITY OF THE DAC TO ZERO, AND THEN TO REACH AN ACCEPTABLE
  FULL SCALE OUTPUT, THE RESULTS OF THE ZERO AND FULL SCALE CHECKS ARE
  THEN COMPARED TO VERIFY THE FULL SCALE RANGE OF THE DAC. THE NEXT
  TEST PROCEEDURE DOES A QUICK CHECK OF THE DAC'S LINEARITY BY BRINGING
  THE DAC'S INPUTS TO HALF SCALE AND VERIFYING THAT THE OUTPUT IS WITHIN
  ONE-HALF LSB OF THE MID POINT OF THE FULL SCALE RANGE. THE LAST TEST
  DOES A "QUICK CHECK" FOR MONOTONACITY. */
                  WRITE 'NOW TESTING DAC (U5)%NL%';


 U5:      USE PMSG='NE5118 DAC',
          USE CMSG='U5';
          /* TEST FOR INTERNAL VOLTAGE REFERENCE ACCURACY */

          /* SET THE POSITIVE SIDE OF THE DC VOLTMETER ON CHA AND THE
           NEGATIVE SIDE OF THE METER ON CHB , THE REFERENCE OUTPUT OF
           THE DAC WILL BE TESTED WITH NO LOAD */

 U5_R:    SET MUX AT(CHA=4:CHB=5);
          SET SCAN AT(CHA=245:CHB=16);
          DELAY 500M;       /* ALLOW FOR DAC REFERENCE SETTLING TIME */
          MEAS DCV INTO VREFOUT HI=5.2 LO=3 8 DLY=30M FAIL(8)[
          WRITE ID=MESFILE'
          DAC U5 FAILED REFERENCE VOLTAGE TEST, EXPECTED 5.0 VDC,%NL%';
          WRITE ID=MESFILE'
          ACTUALLY MEASURED %G% VDC%NL%'VREFOUT; ];

 ENDREF:

 U5_2:    /* FULL SCALE OPERATION CHECK */
          SET MUX AT(CHA=4:CHC=5);
          SET SCAN AT(CHA=181:CHC=16);
          ASSIGN LGC VIHA=5.0 VILA=0.0 VOHA=2.4 VOLA=0.8 LVLA(1-320);

          /* THE FOLLOWING BURST IS USED TO DISABLE ALL BUSSED DEVICES
           AND THE JFET Q2 */
          /* DET DAC TO ZERO */
 U5_Z:    BURST ACTIVE;
          /* DISABLE THE CLOCK */
          $ IC(307,273) IL(307,273);
          /* RESET AND DISABLE THE Z80 */
          $ IC(272,113,146) IL(113,146) IH(272); /* CLAMP RESET AND BUSRQ
                                   */
          LOOP=4 $ IL(272); /* CLOCK THE Z80 FOUR TIMES */
          $ IH(272);
          $ IL(272);
          $ IH(272) END LOOP ;
          $ IL(272);       . /* Z80 NOW RESET */
          $ IC(112,110) IH(112,110); /* TRISTATE THE ROM AND RAMS */
          $ IC(DACDATA,208) IH(208);
          $ IG(DACDATA=B'00000000');
          $ ,
          $ IL(208);
          $ IH(208);

ND BURST ;
```

This procedure is the DAC test for which there is no library procedure. It is inserted in the beginning of stream 110 so that it follows the op-amp test and precedes the power down procedure that turns off power used in the op-amp and DAC tests.

```
        MEAS DCV INTO DACOUT1 HI=39M LO=0.0 DLY=30M FAIL(8) [WRITE ID=MESFILE'
        DAC U5 FAILED TO ZERO, POSSIBLE TRIM PROBLEM%NL%';
        WRITE ID=MESFILE'
        EXPECTED 0.00V ACTUALLY MEASURED %G% VOLTS%NL%'DACOUT1; ],

/* -=-=-=-= SET DAC TO ALL ONES AND CHECK FOR FULL SCALE OUTPUT =-=-=-=*/

U5_F:   BURST ACTIVE;
        /* DISABLE THE CLOCK */
        $ IC(307,273) IL(307,273),
        /* RESET AND DISABLE THE Z80 */
        $ IC(272,113,146) IL(113,146) IH(272); /* CLAMP RESET AND BUSRQ
                         */
        LOOP=4 $ IL(272); /* CLOCK THE Z80 FOUR TIMES */
        $ IH(272);
        $ IL(272);
        $ IH(272) END LOOP ;
        $ IL(272);      /* Z80 NOW RESET */
        $ IC(112,110) IH(112,110); /* TRISTATE THE ROM AND RAMS */

                $ IC(DACDATA,208) IH(208);
                $ IG(DACDATA=B'11111111');
                $ ;
                $ IL(208);
                $ IH(208);
         END BURST ;

        MEAS DCV INTO DACOUT2 HI=10 LO=8.8 DLY=60M FAIL(8) [
        WRITE ID=MESFILE'
        DAC U5 FAILED TO REACH FULL SCALE%NL%'; WRITE ID=MESFILE'
        POSSIBLE OUTPUT OP-AMP PROBLEM OR FAULTY DAC %NL%';
        WRITE ID=MESFILE'
        DAC ONLY REACHED %G% VOLTS%NL%'DACOUT2; ];



/*=-=-=-=-=-=- CHECK FOR THE LINEARITY OF THE DAC -=-=-=-=-=-=-=-=-=*/
        /* HALF SCALE CHECK */

U5_H:   BURST ACTIVE;
        /* DISABLE THE CLOCK */
        $ IC(307,273) IL(307,273);
        /* RESET AND DISABLE THE Z80 */
        $ IC(272,113,146) IL(113,146) IH(272); /* CLAMP RESET AND BUSRQ
                         */
        LOOP=4 $ IL(272); /* CLOCK THE Z80 FOUR TIMES */
        $ IH(272);
        $ IL(272);
        $ IH(272) END LOOP ;
        $ IL(272);      /* Z80 NOW RESET */
        $ IC(112,110) IH(112,110); /* TRISTATE THE ROM AND RAMS */

                $ IC(DACDATA,208) IH(208);
                $ IG(DACDATA=B'10000000');
                $ ;
                $ IL(208),
                $ IH(208);
 END BURST ;

        MEAS DCV INTO DACOUT4 DLY=30M    HI=5.5 LO=4.5 FAIL(8)
         [WRITE ID=MESFILE'
        DAC U5 FAILED HALF SCALE TEST%NL%'; ];

/* =-=-=-=-=-=-=-=- MONOTONACITY CHECK OF THE DAC -=-=-=-=-=-   */


U5_M:   BURST ACTIVE,
        /* DISABLE THE CLOCK */
        $ IC(307,273) IL(307,273);
        /* RESET AND DISABLE THE Z80 */
        $ IC(272,113,146) IL(113,146) IH(272); /* CLAMP RESET AND BUSRQ
                         */
        LOOP=4 $ IL(272); /* CLOCK THE Z80 FOUR TIMES */
        $ IH(272),
        $ IL(272);
        $ IH(272) END LOOP ;
        $ IL(272);      /* Z80 NOW RESET */
        $ IC(112,110) IH(112,110); /* TRISTATE THE ROM AND RAMS */

        /* MONOTONACITY TEST */
                $ IC(DACDATA,208) IH(208),
                $ IG(DACDATA=B'00010000');
                $ ;
                $ IL(208);
                $ IH(208);
 END BURST ;

          MEAS DCV INTO DACOUT1 DLY=30M;
  U5_M2:  BURST ACTIVE;
          /* DISABLE THE CLOCK */
          $ IC(307,273) IL(307,273);
          /* RESET AND DISABLE THE Z80 */
          $ IC(272,113,146) IL(113,146) IH(272); /* CLAMP RESET AND BUSRQ
                           */
          LOOP=4 $ IL(272); /* CLOCK THE Z80 FOUR TIMES */
          $ IH(272);
          $ IL(272);
          $ IH(272) END LOOP ;
          $ IL(272);      /* Z80 NOW RESET */
          $ IC(112,110) IH(112,110); /* TRISTATE THE ROM AND RAMS */

                  $ IC(DACDATA,208) IH(208);
                  $ IG(DACDATA=B'00010001');
                  $ ,
                  $ IL(208);
                  $ IH(208);
   END BURST ;
```

```
        MEAS DCV INTO DACOUT2 DLY=30M;
        LET DACOUT3=DACOUT2-DACOUT1;
        IF DACOUT3 < 0 THEN [WRITE ID=MESFILE'
        DAC U5 NOT MONOTONIC%NL%';
        BITSET (FAIL,8), ];
        IF FAIL(8) THEN [ WRITE ID=MESFILE'
        DAC U5 FAILED ONE OR MORE TESTS%NL%'; ];

U5_13:  CLEAR LGC,
        CLEAR SCAN,
        CLEAR MUX;
        USE NOFSUB PMSG='' DMSG='' CMSG='';

/* =-=-=-=-=-=-=-=-=-=-=-=-=-= END OF THE DAC TEST =-=-=-=-=-=-=-=-=-=-*/

REPLACE DIGINIT    [

FUNDONE:
        CALL PWRDN();

  WRITE 'NOW TURNING DOWN POWER FOR END OF FUNCTIONAL TESTING%NL%';

ADONE:  USE NOFSUB PMSG='' DMSG='' CMSG='';
        CLEAR SCAN;
        CLEAR MUX;

        WRITE 'NOW TURNING UP POWER FOR DIGITAL TESTING%NL%';

        ON ABORT CALL PWRDN();
PWRUP:  SET PIO(0) HRLY(CLOSE 3);        /* GROUND PCB */
        SET PIO(0) HRLY(CLOSE 6); /* CONNECT FIXED P.S. TO DUT */
        DELAY 10M;
        SET PIO(0) RLY(CLOSE 12); /* TURN ON FIXED P.S. OUTPUTS */
        DELAY 10M;

                WRITE 'NOW TESTING NON BUSSED ICS%NL%';
    ]

INCLUDE ALLBURST        [

$ IC(302) IL(302);

]

REPLACE DIGEND        [

DDONE:
        CALL PWRDN();

]

INCLUDE AFTER (250)  [

 SUBROUTINE FAILCOMP();
        WRITE ID=MESFILE ' %037%%S%%036% FAILED %S% %S%
'        FLABEL,FTYPE,FSTAT;
        WRITE ID=MESFILE '%S%%NL%' FDMSG;
        WRITE ID=MESFILE 'MEASURED VALUE = %5G%%035%%NL%' FMEAS;

        IF FPMSG<>'' THEN [WRITE ID=MESFILE '%S% %NL%' FPMSG;
                         WRITE ID=LOGFILE '=%S%' FPMSG; ];
 END SUBROUTINE FAILCOMP;

                SUBROUTINE FAILCRLED();
                IF FAIL(5) THEN [
                 WRITE ID=MESFILE '
                 * DCI IN COMPLIANCE *
                %037%%S%%036% MAYBE OPEN OR BACKWARD %NL%'FLABEL;
        ];
                 ELSE [
                 WRITE ID=MESFILE '%037%%S%%036% FAILED %S%%NL%
                 'FLABEL,FSTAT;
                 WRITE ID=MESFILE ' %S% MEASURED %5G% VOLTS %NL%
                'FDMSG,FMEAS; ];
                IF FPMSG<>'' THEN WRITE ID=MESFILE ' %S%%NL% ' FPMSG;

        END SUBROUTINE FAILCRLED;


SUBROUTINE PWRDN();

PWRDN:  SET PIO(0) RLY (OPEN 12);        /*TURN OFF FIXED P.S. */
        DELAY 10M;
        SET PIO(0) HRLY (OPEN 6);        /* DISCONNECT FIXED P.S. FROM DUT
         */
        DELAY 10M;
        SET PIO(0) HRLY (OPEN 3);        /* GROUND RELAY */

END SUBROUTINE PWRDN;

]
```

Circuit Description Preparation – Section 6

6.1 INTRODUCTION

The first step in generating a test program, using the automatic test generator (ATG) process, is to write a circuit description source (.CKT) file from the circuit schematic and parts data. The .CKT file is made up of high-level Circuit Description Language (CDL) statements. These statements can be written manually using an editor, or you can run the .CKT generator (CKTGEN) routine. Since it saves time, and can be used effectively by a new user, the .CKT file is normally created by using CKTGEN. An application for creating a .CKT file manually and not using CKTGEN is when circuit description information already exists in some form, usually originating from a CAD system database. Often, a CAD database can be edited to create a .CKT file faster than one can be built by starting from scratch with CKTGEN. In this event, however, a knowledge of the CDL language and syntax is a prerequisite.

The circuit description provides ATG with complete component and interconnection information on the pc board type to be tested. Typically, the user traces through the circuit diagrams, assembling the circuit data needed by ATG, and codes this information in the .CKT circuit description program as described in this section. Regardless of the method used to generate the .CKT file, ATG checks it for syntax and connection errors, then compiles it for use in generating the test program.

CKTGEN is a comprehensive, interactive mode containing several commands, much like an editor. Yet only a few of these commands are needed for simple circuit data entry. The advanced user has many more commands available to accomodate special devices and circuit configurations.

6.2 SECTION ORGANIZATION

This section is structured to permit the new user to begin circuit description entry on an elementary level. The features and capabilities for the more experienced user are then presented. A summary of the CKTGEN commands and paragraphs describing related activities including the circuit file reader and the command description language (CDL) are also contained in this section.

6.3 CKTGEN BLOCK DIAGRAM DESCRIPTION

Figure 6-1 is a block diagram showing the CKTGEN modes of operation. As illustrated, the major output file is the .CKT file that is used as an input file by the CDL compiler in ATG. Additionally, there are a few status report files (.CRP,.ERC) and a nail database file (.NDB) used as an input file in the nail assignment mode. Figure 6-1 shows that a .CIR database file is produced using the Create mode. A .CKT circuit description file is then made in the Generate mode. Revise mode accommodates changes to an existing .CIR database file. Finally, the Read mode accepts an existing .CKT file and produces a .CIR file that can be modified. This mode is used when a .CIR does not exist (hasn't been kept).

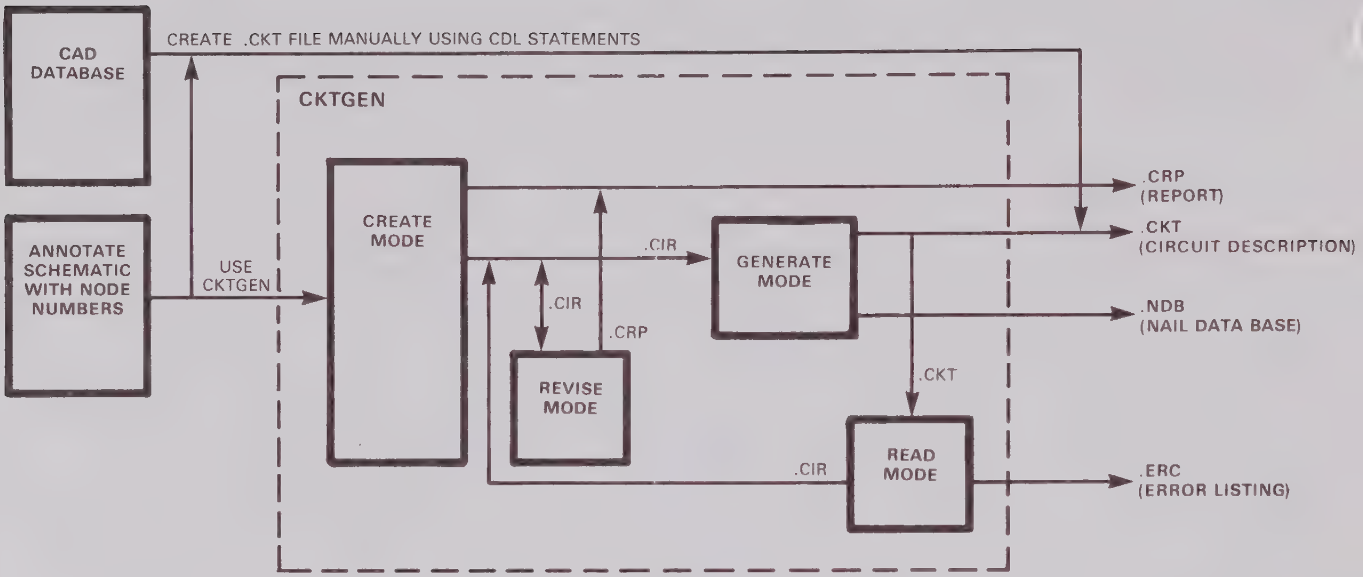

CAD DATABASE
CREATE .CKT FILE MANUALLY USING CDL STATEMENTS
CKTGEN
ANNOTATE SCHEMATIC WITH NODE NUMBERS
USE CKTGEN
CREATE MODE
.CIR
.CIR
.CRP
GENERATE MODE
REVISE MODE
.CKT
.CIR
READ MODE
.CRP (REPORT)
.CKT (CIRCUIT DESCRIPTION)
.NDB (NAIL DATA BASE)
.ERC (ERROR LISTING)

Figure 6-1. CKTGEN Block Diagram.

6.4 PREPARING FOR CKTGEN

A schematic that agrees with the design of the known-good boards must be provided. It should be a film positive for ease of marking and copying.

The programmer should mark each node (common circuit point) with a unique node name (e.g., N1, N2, etc). If these node names have a numerical value, then the node names can be considered as temporary nail numbers which can be easily cross referenced to permanent nail numbers produced by nail assignment. Nail Assignment converts 2271, 2272 or 2275 temporary nail numbers to actual nail numbers in the .TPG. The Nail Assignment mode does not apply to the 2270, where the assigned numbers are the permanent numbers. A suggested method of marking for 2271, 2272 and 2275 systems is to write on the schematic diagram, next to each node, the temporary nail number followed by a slash (/) and a space. The space is filled-in later with permanent nail numbers.

A parts list should provide the value and tolerance of all component characteristics so each component can be accurately described in CKTGEN. Commercial type numbers of all ICs should be provided so that the proper library procedure can be used or so that a special test can be written if a test is not in the library. Data sheets and circuit descriptions should be provided for custom components to allow a special functional test, an orientation test, or an analog component description to be written.

6.5 CKTGEN FOR THE NEW USER

6.5.1 General

Using CKTGEN to establish a circuit description basically consists of filling in the blanks of several menu-like pages. There will be a page for each component. For each component you will specify:

type	(component type, e.g., R for resistor, 7400 for a NAND gate package)
name	(component name, e.g., R1, U17, etc)
connections	(usually the temporary node number assigned to each component pin),
flagspecs	(special characteristics depending on the component),
messages	(to be printed later by the test program)

When all the components have been entered, a .CKT file can be generated. This file and a few others are required by ATG as input files to produce a working test program.

The training board supplied as an option with the system is used as an example. A .CKT file for this board and the schematic are included in paragraph 6.11. It is suggested that the new user practice and learn CKTGEN by entering at least part of this circuit description as described in the following paragraphs. The fastest way to become familiar with CKTGEN is to step through the following constructs before reading CKTGEN in detail. This will provide you with an understanding of the concepts before you are exposed to the full set of commands and features. This should be done even though some of the prompts or features might not be understood at this time. The CKTGEN monitor page, the BOARD page, the Circuit Description Language, the command set and other aspects of CKTGEN can then be read with a better understanding. After first completing the simple exercise, read through the CKTGEN commands (para 6.8) to become aware of the full set of commands and their abbreviated forms. You will then be able to contribute to the building of a circuit-description database for even the most complex of circuit boards.

6.5.2 Introduction

The following example will introduce the concepts and structure of CKTGEN. For this description, a familiarity with the basic 227X monitor operations (including logon/off and HELP) is assumed.

CKTGEN prompts the user to input connections and value information for each component. CKTGEN uses this information to produce a data base (.CIR) file, from which it then automatically builds the .CKT circuit description file for ATG.

Before performing the example, spend a few minutes reading the next few paragraphs about the structure of the HELP available.

6.5.3 CKTGEN Help Utility

Extensive support within CKTGEN is available to aid you in understanding the command set and definitions of the options and flagspecs. There are a few levels of HELP in a utility routine. If, for example, you are filling out the CKTGEN BOARD page and you are unsure of the intent of a prompt, press the HELP (PF4) key before responding to the prompt. A brief explanation of whatever is being prompted for will then be displayed. A more detailed explanation can also be displayed if the brief explanation is insufficient.

The CKTGEN Help Utility provides 2 menus of topics for which help is available. These are the COMMAND menu and the OPTIONS menu. Each item on the menu has a page of information.

Initially, HELP is accessed by entering one of the following commands from anywhere within CKTGEN. These commands are terminated by entering a carriage return <RETURN>. Note that a help command must be preceded by the # symbol.

#HELP	invokes the CKTGEN HELP Utility and describes the help facility.
#HELP OPTIONS	displays a menu of CKTGEN options. This menu is listed below.
#HELP (option name)	displays the help page for the named option. (The parentheses are not typed, e.g., HELP PRESET displays a description of the potentiometer flagspec PRESET.)
#HELP COMMANDS	displays a menu of CKTGEN commands.
#HELP (command name)	displays the help page for the named CKTGEN command. (The parentheses are not typed, e.g., HELP DENOTE displays a description of the DENOTE command.)

At the end of each help page is the prompt 'Enter topic, 'Commands,' or 'Options.' Press <RETURN> to return to the CKTGEN page you were on when you requested help.

The HELP OPTIONS menu includes the following options. To get general help about CKTGEN type GENERAL from the HELP OPTIONS page.

BOARD FLAGSPECS
CERR
CHIRESID
CLOCK DEFAULT
CLKFREQ
CLORESID
COMPONENT FLAGSPECS
DCR
DEVICE
DR
DRH
DRL
FAST NAIL COUNT
FAST ATG
FREQ
FVONLY
GENERATE
HIGH-SPEED
HIGHEST DIGITAL NAIL
HYBRID NAIL COUNT
IC
IDR
IFV
IGT
ILIM
INPIN
IZT
KNEE
LERR
LHIRESID
LLORESID
LOWEST DIGITAL NAIL
MAXIMUM ANALOG NAIL
MSG
NAME
NEGRAIL
NOFAST
NOTEST
OUTPIN
PINCOUNT
POSRAIL
PRESET
PROBE
R23
RERR
RHIRESID
RHIV
RLORESID
RLOV
RTOL
SYNC
TESTFREQ
TOLERANCE
TYPE
VA
VALUE
VBE
VCE
VDIFF
ZZT

The HELP COMMANDS menu includes the following COMMANDS. To get general help about CKTGEN type GENERAL from the HELP COMMANDS page.

ADD
ARROWS
BOARD
CAP
CONFIRM
CON CHA
DEFINE
DELETE
DENOTE
ERASE
EXIT
EXPAND
GENERATE
GOTO
HELP
INSERT
MORE
MOVE
NEWNAME
NEWTYPE
NOCONFIRM
NOEXPAND
NOPINSALL
PINSALL
QUIT
REMOVE
REPORT
SAME
SHOW
TARGET
UNCAP
VISIT

6.5.4 CKTGEN - An Example

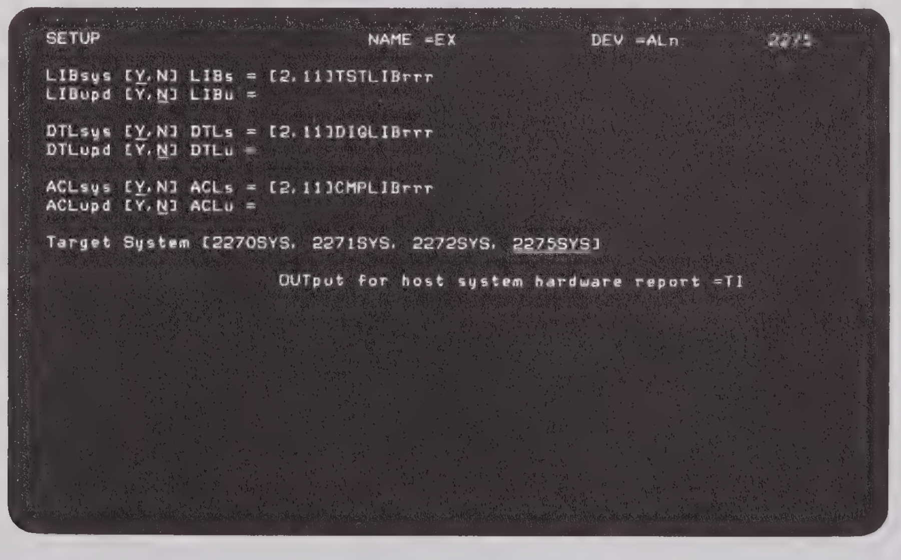
The following paragraphs describe how a circuit description (.CKT file) is created using the node and parts data for this board (paragraph 6.11). Only a few of the components will be entered in this exercise, first to show you how to create a database file and then generate a .CKT circuit description file.

To prepare for CKTGEN, temporary node numbers/names are assigned to each circuit node. Typically, the nodes are assigned numbers, e.g., N25 or N152, etc. These numbers are considered temporary and may not completely correspond to the final numbers defined after nail assignment and CKTGEN nail allocation aspects have been considered. The final assignments depend on system type, pin count, numbers of pin types available and other factors. You will not need to do this for the following exercise, however.

To start, turn-on and ready the system or terminal. Initially, the 227X DIAGNOSE monitor page is entered. The options can be left in their default (left) positions.

SETUP Mode

Next, proceed to the Setup Page by entering a #SET command and enter the setup page (Figure 6-2). Check that the name and device specifications are appropriate. For this example the NAME=EX is used, and the device is typically ALn: where n is a digit specifying the disk drive where the database will be built. Next, check the library specifications to be sure they are appropriate. These specifications vary between systems and users. The one shown is typical, where rrr is the software release/version number. The system manager can furnish these file specifications if they are not already filled in. Finally, check that the target system (2275 in this example) is selected.



```
SETUP                        NAME =EX               DEV =ALn        2275

LIBsys [Y,N] LIBs = [2,11]TSTLIBrrr
LIBupd [Y,N] LIBu =

DTLsys [Y,N] DTLs = [2,11]DIGLIBrrr
DTLupd [Y,N] DTLu =

ACLsys [Y,N] ACLs = [2,11]CMPLIBrrr
ACLupd [Y,N] ACLu =

Target System [2270SYS, 2271SYS, 2272SYS, 2275SYS]

                     OUTput for host system hardware report =TI
```

Figure 6-2. CKTGEN Example (Setup).

When you have completed this, perform a #CK command to enter CKTGEN.

CKTGEN Monitor Page

Select the following options:

```
Mode [CREATE] (A new file is going to be made.  This can be specified from
          the bottom of the page by entering #MODE=CREATE .)
DTLusr [N] (No user digital library is going to be used)
```

Then, press the <RUN> or <LINEFEED> key. This moves the cursor to the bottom of the page and performs the command #RUN automatically to enter the BOARD page.

BOARD Page

The board page options are specified by answering the questions as they are prompted for at the bottom of the screen, or the options can be filled in from the command line in the same manner as in the monitor. The questions and the responses are as follows:

> **NOTE**
>
> To return to a prompt that is skipped by mistake, or to change the response, cycle back to it by using the up/down arrows or pressing the <RETURN> key until the prompt re-appears. (Note also the use of the command-mode commands described in paragraph 6.9.)

```
ENTER TARGET 2270 [N,Y] or #: N
ENTER TARGET 2271 [N,Y] or #: N
ENTER TARGET 2272 [N,Y] or #: N
ENTER TARGET 2275 [N,Y] or #: Y (already Y from SETUP page -
                                 press <RETURN>)
ENTER Hybrid nail count or #: 256      (arbitrary for this example)
```

Press <RETURN> for the following options:

```
ENTER Probe or #:
ENTER Sync or #:
ENTER Board Clock FREQ or #:
ENTER Clock Default [CD,CS] or #:
ENTER BOARD FLAGSPECS
ENTER COMPONENT TYPE FLAGSPECS

ENTER TARGETS complete [N,Y] or #: Y    (Enter a Y to this prompt)
```

You will not be able to leave the board page (even for HELP or to QUIT) until the data for all active targets have been specified.

Entering EXTernals

The EXTernals (power and ground connections) are usually entered first, followed by the analog components, then the digital IC's. The power EXTernals VCC and GND have special names as described in the Circuit Description Language at the end of this section. The component values and node names used in this exercise can be read from the schematic and the .CKT file. Press <RETURN> to terminate each entry.

```
ENTER Type of component or #: EXT    (It is posted in the upper-left part of
                                       the screen)
ENTER Name of component or #: GND
ENTER [1] or #:16   (note that when the number 16 is entered, it is auto-
                    matically prefaced by an N when it is posted in the
                    NODE CONNECTIONS area of the screen)
```

Press <RETURN> until the prompt for the next component type appears. (Alternately, enter #NAME=VCC.) The first external is saved and now a new EXTernal component page is made for the +5-V external connection.

```
ENTER[1] or #:15
```

The 2 remaining externals VCC15 and VCC15N are entered in a similar manner (#NAME=VCC15 with connection 17, followed by a #NAME=VCC15N with connection 18).

Entering Analog Components

Now, since there are no more externals to enter, we can begin entering capacitors. Begin for C1 by bringing up a component page for tantalum capacitors by entering #TYPE=CP. (The symbol CP specifies polarized tantalum capacitors. A table in the Circuit Description Language lists all of the symbols.) Then fill in the page as follows:

```
ENTER Name of component or #: C1
ENTER Value or #: 10U
ENTER Tolerance or #: 20
ENTER[P] or #:15
ENTER[N] or #:309
```

FLAGSPECS are additional optional specifications that vary depending on the component. No FLAGSPECS are entered for this component. Proceed to the next component by entering #TYPE=C. The following capacitors are not tantalum (CP) types. (Commands like the #TYPE command can be entered at any point as long as they are preceded by a # symbol as described in paragraph 6.9) The C1 data is recorded and the type is changed from CP to C. Now, with a new blank type C component page enter the data for C2 in the same manner as with C1 i.e.,

```
ENTER Name of component or #: C2 ENTER Value or #: 10.0N ENTER Tolerance or #:
20 ENTER[1] or #:53 ENTER[2] or #:18
```

Then, since C5 is of the same component type (C), and is of the same value and tolerance, perform a #SAME=C5 command and enter the connections 83 and 16. The remaining analog components (including diodes and transistors) could be entered in similar fashion. This is done later to obtain a complete circuit description .CKT file for the training board, but is beyond the scope of this exercise.

Entering the Digital Components

Digital components can be entered the same way analog components are entered. For example, perform the command #TYPE=7400. The type 7400 IC is recognized by the digital library. A blank digital component page is then displayed. The library then automatically furnishes the pin count (14 for a

7400) and the pin names for this IC. CKTGEN then prompts for the device name. Enter U1. The node connections are then entered one at a time as prompted for:

```
[1]301  [9]212
[2]302  [10]111
[3]303  [11]112
[4]303  [12]110
[5]304  [13]111
[6]305  [14]16
[7]15
[8]110
```

After these connections are entered, perform the command >1. The cursor and prompt are moved (GOTO command) to pin 1 of this IC.

The NODE EXPANSION section of the display then shows the expansion for pin 1 (U1.1) with 3 lines of information as follows:

```
----------------------------NODE EXPANSION----------------------------

node: N301 U1.1
pin:     [A      I]
nails:(H301)
```

This display shows that U1 pin 1 is an IC input named A, and that hybrid pin 301 has been assigned. The parentheses indicate that this is a tentative assignment and subject to later change in ATG.

Other digital components can be entered in the same manner as this elementary component U1 as long as the component is known to the digital library (system, update, or user). If the component is unknown, it must first be defined. Then node connections to it can be added as you have just done for U1.

Note that the keyboard arrows can be used to move from one pin to another. For example, the right arrow can be pressed from pin 1 to move to and prompt for pin 9. The up/down-arrows also function in a similar way.

There is a line at the bottom of the screen stating:

```
stats <ic,bd,max>: H=11,16,256
```

This indicates that 11 hybrid pins have been assigned to this IC, 16 have been currently assigned to the board, and that a maximum of 256 can be allocated as specified on the CKTGEN Board page.

Enter the command #HELP=COMMANDS. (Type the # symbol also.) A menu of commands for which help is available is displayed. You can then enter a command eg ADD. To return to U1 press <RETURN>. Another help menu can be accessed by entering #HELP=OPTIONS. Press <RETURN> after help for any options is displayed.

To terminate this session enter #EXIT. A status report (the .CRP) file is displayed. Control then returns to the CKTGEN monitor page. A #SHU command can then be performed to terminate operation. To continue later, enter the CKTGEN monitor page, and perform the command #MODE=REVISE.

When an entire board has been entered in the database, a circuit description (.CKT file) can be made from the CKTGEN monitor page with the command #MODE=GEN (for generate). Then press <RUN> or <LINEFEED> to run.

6.6 CKTGEN PAGE DESCRIPTIONS

6.6.1 General

CKTGEN has the following different types of pages which are described briefly below and in detail with illustrations in the following paragraphs.

CKTGEN Monitor	to select mode of operation (create, read, revise, generate), and file specifications for the associated data.
BOARD	to specify the target system type(s) for which the circuit description will ultimately be used. Tester-specific pin and nail information is also specified for each target selected.
Component Type	to specify each component on the board. Each component type has its own page format.
HELP	to display information on how to use or interpret some aspect of CKTGEN.
REPORT	to display among other information, nail and pin allocation and capacity status.
DEFINE TYPE	to allow the user to define IC information for a component that does not exist in any library. If a component does exist in a library, a DEFINE TYPE command will display the component, but not allow it to be changed (unless it is located in a user library).

6.6.2 CKTGEN Monitor Page

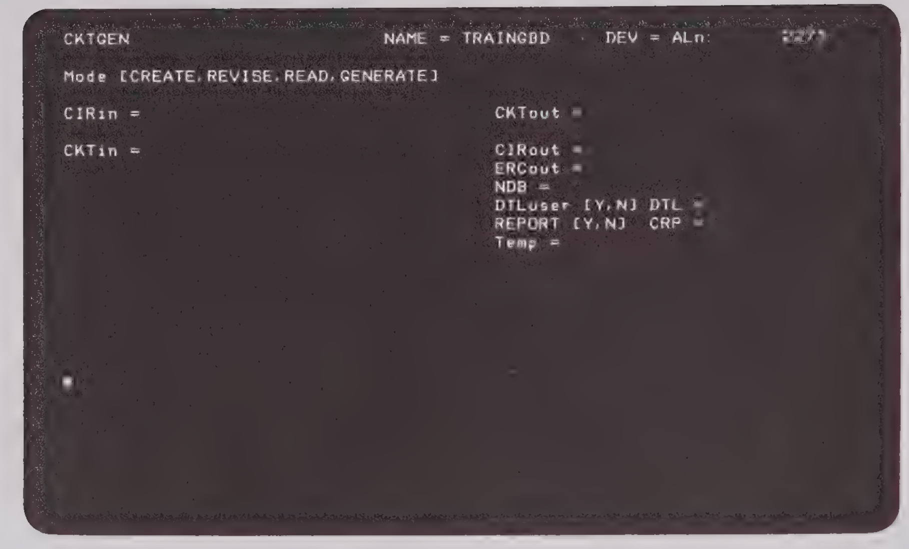
The CKTGEN monitor page, shown in Figure 6-3, is displayed when CK (or CKTGEN) is typed from the monitor prompt (#). The command line method of accessing CKTGEN would be to type the following command after the prompt (#) on any page:

```
#CK SY:PCBI
```

for NAME = PCBI and DEV = SY:.

NAME =
Specifies the file name default for the .CIR and .CKT files that are created. (The .CIR file is the database file produced in the CREATE mode described below.) The default can be overridden by entering a different name in the "CIR =" and "CKT =" file spec options.

DEV =
Specifies the default device for the .CIR and .CKT files.



```
CKTGEN                             NAME = TRAINGBD     DEV = ALn:

Mode [CREATE,REVISE,READ,GENERATE]

CIRin =                                       CKTout =

CKTin =                                       CIRout =
                                              ERCout =
                                              NDB =
                                              DTLuser [Y,N] DTL =
                                              REPORT [Y,N]  CRP =
                                              Temp =
```

Figure 6-3. CKTGEN Monitor Page.

Mode [CREATE,REVISE,READ,GENERATE]

These options let you choose to:

create a new database (.CIR) file
revise an existing .CIR file
read an existing .CKT file and produce a .CIR file from it
generate a new circuit description (.CKT) file.

[CREATE] Opens a new database file with the filename specified by "NAME =" and extension .CIR on the device specified by "DEV =". If a file by this name already exists on the specified device, a warning will be displayed on the screen and REVISE will be enabled.

[REVISE] Reopens an existing .CIR database file with the filename specified by "NAME =" with the extension .CIR on the specified by "DEV =". If a file by this name does not exist on the specified device, CREATE will be enabled.

If a new target (% adaptor section) is to be added to a .CIR file, it is accomplished with a VISIT BOARD command.

[READ] Allows you to read an existing .CKT file and create a .CIR file from it, with an error listing to the .ERC file. A circuit read is performed when there is no existing .CIR that can be modified using CKTGEN, but a .CKT has been previously entered manually, or the .CIR was discarded earlier. Refer also to the Circuit File Reader description, paragraph 6.10.

[GENERATE] Creates a new circuit description file with filename specified by the "NAME =" and extension .CKT on the device specified by "DEV =". If a file by this name already exists on the specified device, a new version is created.

CIR =

Allows you to specify that the database file be named or located other than what "NAME =" and "DEV =" specify. This is also useful if you want to GENERATE from an old version of the .CIR file. Refer also to the following note.

CKT =

Allows you to specify that the circuit description file that CKTGEN generates will be named or located other than what "NAME =" and "DEV =" specify. Refer also to the following note.

NOTE

Figure 6-3 shows 2 sets of CIR = and CKT = specifications. One of these will be suppressed. During a READ mode, CKT is the input and CIR is the output; for other modes the opposite is true. The specifications for CIR and CKT are the same for both sets, i.e., entering the CKT (CIR) name for the READ mode will also affect the CKT (CIR) name for the other modes.

ERCout =

Any error messages generated during a READ mode (circuit file reader) operation are stored in the file designated by the ERCout = specification.

NDB =

CKTGEN will create an .NDB file if it finds any active multiplexed testers with target adaptor sections when GENERATE mode is run. The .NDB file will consist of a %NAILMAP ECO section for each active multiplexed tester with a target (nontemporary) adaptor section containing non-fixed nails when GENERATE is run. When a .CIR file is created or revised and nail changes are made, this .NDB file will show these nail changes. The .NDB file is used as an input file in the Nail Assignment (ECO) mode.

DTLusr [Y,N] DTL =

The user digital library will be used by CKTGEN to display digital IC information, and for the insertion or modification of a CKTGEN test skeleton for an IC that is not in any digital library or was previously entered into the library via CKTGEN. This option is set separately from the same question on the ATG monitor page. Changing one of these does not change the other. Specifying a user library file here does not create one if no file currently exists. Perform a #DTL command from the top level of CKTGEN to enter the DTL mode.

REPORT [Y,N] CRP =

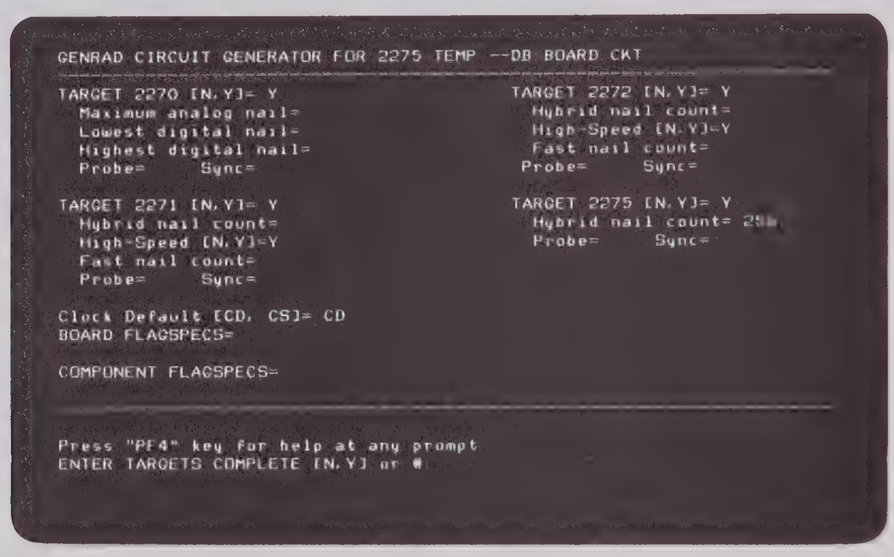
For the CREATE and REVISE modes, if this option is set to Y, the CKTGEN REPORT page is written to the .CRP file. The REPORT page contains statistics on the types of nails allocated, test skeletons defined, nodes having no pins, etc. This page is displayed when a REPORT command is issued. The report file is described in paragraph 6.6.5

TEMP =

This is a scratch file specification that will be used for temporary storage of any newly-defined IC's until they are placed in the user's digital library. (The actual transfer to the library from this temporary scratch file occurs after an INSERT command is performed followed by an exit from CKTGEN).

6.6.3 Board Page

The BOARD page is entered when the options on the first page of CKTGEN have been selected, and a RUN command has been performed (see Figure 6-4). Any target systems other than the one specified in the SETUP mode must be specified before circuit description data is entered. You can run CKTGEN from a system or programming station that is different from the one selected in the SETUP mode. Or, you can compile a circuit description for more than one type of tester simultaneously. Since nails are allocated differently between testers (even the same types of testers with different quantities of pin types), the information requested by the BOARD page specifies the characteristics of each tester for which the resulting .CKT file will be used.



```
GENRAD CIRCUIT GENERATOR FOR 2275 TEMP --DB BOARD CKT
------------------------------------------------------------------------------
TARGET 2270 [N,Y]= Y                       TARGET 2272 [N,Y]= Y
  Maximum analog nail=                       Hybrid nail count=
  Lowest digital nail=                       High-Speed [N,Y]=Y
  Highest digital nail=                      Fast nail count=
  Probe=     Sync=                          Probe=     Sync=

TARGET 2271 [N,Y]= Y                       TARGET 2275 [N,Y]= Y
  Hybrid nail count=                         Hybrid nail count= 25
  High-Speed [N,Y]=Y                         Probe=     Sync=
  Fast nail count=
  Probe=     Sync=

Clock Default [CD, CS]= CD
BOARD FLAGSPECS=

COMPONENT FLAGSPECS=

------------------------------------------------------------------------------

Press "PF4" key for help at any prompt
ENTER TARGETS COMPLETE [N,Y] or #
```

Figure 6-4. CKTGEN Board Page.

The header section of the BOARD page specifies the target system type as selected on the SETUP mode page. This system type is also the one for which node expansion is shown later when components are entered. (The tester specified can be changed using a #TARGET= command.)

The BOARD page options are selected by responding to the prompts as they appear one-by-one at the bottom of the display. If a response to a system type is Yes, you are prompted for nail capacity and other tester-specific information. If a response to a system type is No, you will be prompted for the next system type. No prompt for the next system type occurs if the information for the current type is either incomplete or incorrect. The default for the target sytem type stated in the header section of this page has the default Yes already selected and is the first target that is prompted.

2270 Target

When the 2270 is selected as a target, the following 2270 prompts appear:

Maximum analog nail =
Allows you to specify the maximum number of analog nails that will be used. This helps prevent you from entering a nonexistent nail into the circuit description. (Since the first 32 nails must be analog or hybrid, this number must be at least 32.)

Lowest digital nail =
Allows you to specify the first digital nail. This helps prevent you from entering a nonexistent nail into the circuit description.

Highest digital nail =
Allows you to specify the last digital nail. Again, this helps prevent you from entering nonexistent nails into the circuit description.

Probe=
Allows you to specify a dedicated test nail connected to the PROBE connector on the front panel of the tester.

Sync=
Allows you to specify a dedicated test nail connected to the SYNC connector on the front panel of the tester.

2271/2272 Target

When a 2271 or 2272 is selected as a target, the following prompts appear:

Hybrid nail count or #:=
Allows you to specify the number of 2271 (2272) system test-fixture nails assigned or used in the circuit description file.

High-speed [N,Y] or #:
If Yes is selected, the following prompt for fast nail count appears.

Fast Nail Count=
Enter the number of nails available that can be used for high-speed testing.

Probe= Sync=
These options are the same as for the 2270 tester described above.

2275 Target

When a 2275 is selected as a target the options are the same as for the 2271 and 2272 except that there is no High-speed [N,Y] option, therefore this prompt and the associated Fast Nail Count= prompt are suppressed.

After the target information consisting of system-specific information has been entered for one or more testers of interest, you are prompted for the following test-specific information:

Clock Default [CD, CS]=	This sets the clock default to either clock driver (CD) or clock sensor (CS), and is associated with defining an IC (para 6.7.4).
BOARD FLAGSPECS=	Flagspecs that are to be used for all tests on the board are specified here by entering BOARD=flagspec1,flagspec2, etc. Each flagspec must be separated from the next one by a comma. The BOARD= line can be up to 72 characters (max). The #ERASE command can be used here.
COMPONENT FLAGSPECS=	Flagspecs entered here apply to specified component types using any selected target. Each flagspec must be separated from the next one by a comma. A semicolon can be used to gang flagspecs together for more than one component type. For example, to specify certain flagspecs for resistors and then a component such as a type 7400, the line could read: R=RERR1, RLORESID=2; 7400=NOFAST Note that the 2 flagspecs associated with resistors are separated by a comma, while a semicolon separates the 2 component types. The COMPONENT FLAGSPECS= line can be up to 72 characters (max). The #ERASE command can be used here.

The first prompt to appear on this page will be the first incomplete target or the TARGETS COMPLETE [N,Y]: prompt. The answer to TARGETS COMPLETE must be Yes to proceed to the components page.

6.6.4 Component Pages and Entering Data

Entering Circuit Information

CKTGEN prompts you for all of the circuit description information for a given component. Each page includes the appropriate prompts for a component type (R, C, L, etc). As you respond to the prompts, your responses will be recorded on the page display. Within each page display, the response records are arranged in sections:

component identification and value section
component connections section
flagspec selections section
node expansion section
prompt section

There are several ways to position the cursor to enter information into the database. The following include the elementary ways, but note the CKTGEN command mode set (paragraph 6.9) also:

1. Press the <RETURN> key until desired prompt appears. Then type the appropriate information.

2. Type #prompt = information after any prompt. This assigns the information for the selected option and sets CKTGEN's ENTER: prompt to the option immediately following the selected one, e.g., #TOLERANCE=1.0. Alternately, perform one of the command mode commands to position the cursor (para 6.9).

3. Use the up/down and right/left arrow keys to move the cursor around within the page. The right/left arrows work only in the header, node connections, and flagspec areas of the display.

Component Identification and Value

When CKTGEN has been entered, the system prompts the user to ENTER COMPONENT TYPE or #:. To respond to the prompt, enter one of the ATG symbol component types, (e.g., R for a resistor, QN for an npn-transistor, etc); press <RETURN>. The system displays the CKTGEN Component Type Display for the component symbol you selected. The available component types are listed in Table 6-1, along with corresponding sample displays found in Figures 6-5 through 6-19.

Table 6-1
LIST OF AVAILABLE COMPONENT TYPES

Component Type	Abbreviation	Sample Displays
External Connection	EXT	Figure 6-5
Resistor	R	Figure 6-6
Capacitor	C	Figure 6-7
Polarized Capacitor	CP,CP1	Figure 6-8
Inductor	L	Figure 6-9
Potentiometer	RV	Figure 6-10
Jumper	J	Figure 6-11
Fuse	F	Figure 6-12
Diode	CR	Figure 6-13
Transistor	QN, QP	Figure 6-14
Zener Diode	VZ	Figure 6-15
Digital Component (Integrated Circuit)	IC,DIC	Figure 6-16
Analog Component	AIC,IC	Figure 6-17
Silicon-Controlled Rectifier	SCR	Figure 6-18
Field-Effect Transistor N- and P- Junction	NJFET,PJFET	Figure 6-19

After a component type has been entered, the proper page is displayed and you will be prompted for the component's name and then its value and tolerance, if these are appropriate, after which you enter a response or press <RETURN>.

If an unknown component type is specified, the system prompts the user as follows:

```
ENTER ANALOG OR DIGITAL (A/D) or #:
```

Type A if the component is an analog type, or D if the component is a digital type. The system will respond by displaying either the analog component page (see Figure 6-17) or the digital component page (see Figure 6-16).

Once an A or D type page is specified, the system will default to that page for the rest of the session, unless explicitly changed by the user. To change between the analog component page and digital component page, type

```
#DEVICE = ANALOG

        or

#DEVICE = DIGITAL
```

Initially, when the first unknown component type is specified, the system defaults to the digital page. At that time, the user's response to the question "ENTER ANALOG OR DIGITAL (A/D) or #:" determines the subsequent default. From then on, the user can change the default, as desired, by typing the command #DEVICE = ANALOG or #DEVICE = DIGITAL. (The short form is #DEV=A or #DEV=D.)

Connections

After describing the component, its connections must be defined. There are 3 methods of entering connection information for a component:

1. Enter connection information, using the TEMPORARY NAIL number assigned to a component lead (if there is one).

2. Enter connections, using OTHER COMPONENT CONNECTIONS. It is possible to enter component connections with another component's connections, provided the other component's connections have already been entered.

3. Enter connections, using an EXTERNAL NAME. It is possible to enter component connections using external names, provided the external name has already been entered. It is important that the external name be connected already, otherwise the .CIR file will contain a node and an external with the same name but having no common connection. This will result in a subsequent CDL error. For example, the external name GND can be entered for that connection with any component instead of the nail assigned to the ground node if ground has been previously entered as an external (TYPE=EXT).

Flagspecs

After the connections of a component are specified, you will be prompted for any appropriate flagspecs. To bypass any flagspec prompt which you don't wish to specify, press <RETURN>.

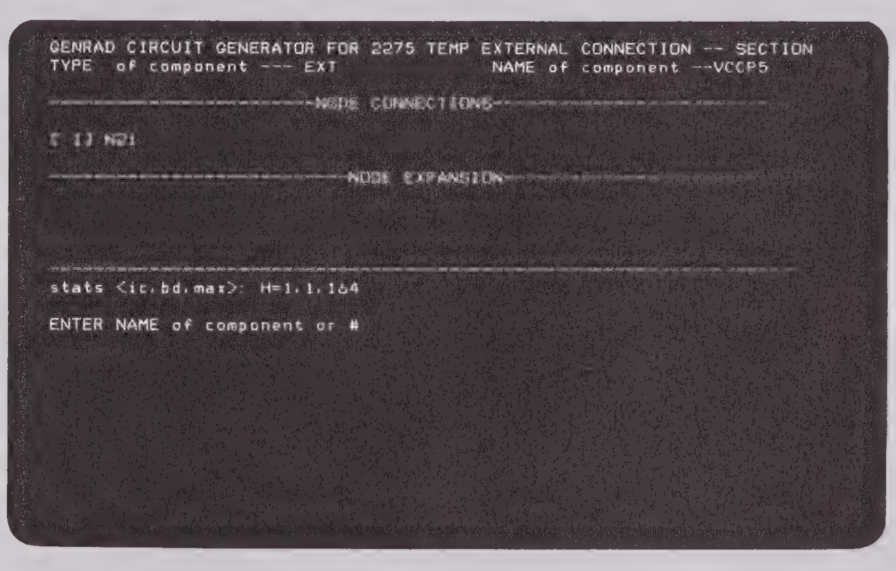
After entering all the information for the component named, you must do 1 of the following:

If there are other identical or nearly identical components to be entered, type #SAME=newname (where newname is the name of the next component to be entered). You will then be prompted for the node connections which are often nearly the same.

If the next component to be entered is of the same type as the one just entered, type #NAM=newname (where "newname" is the name of the next component) to save the previous component data and to clear the display.

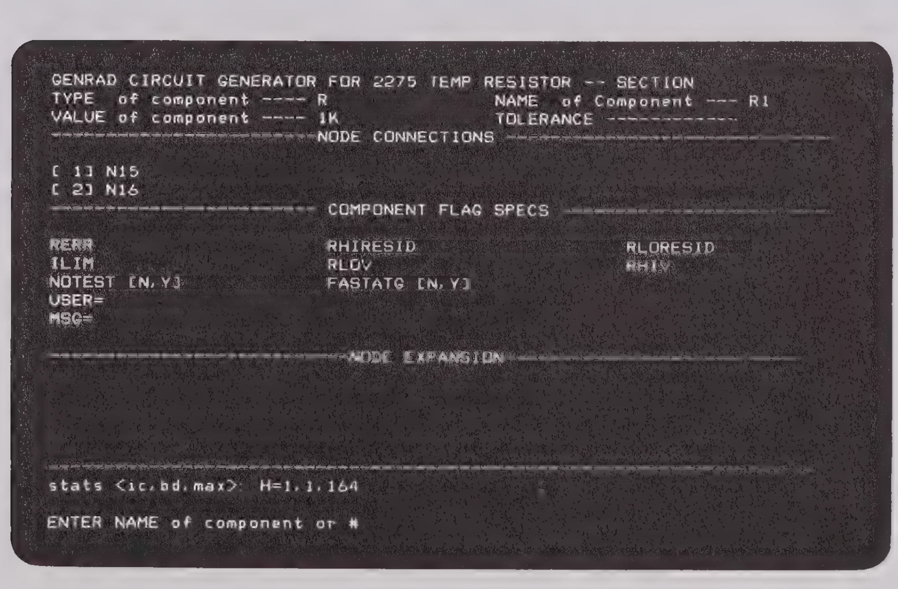
If the next component to be entered is of a different type than the type just entered, type #TYP=newtype (where "newtype" is the ATG symbol for the next component type) to save the previous component data; the newtype CKTGEN display is shown. Enter the new component name and information, as described previously.

If the component that was entered is the last one on the board, type #EXIT to save the component data and <RETURN> to the CKTGEN monitor page.



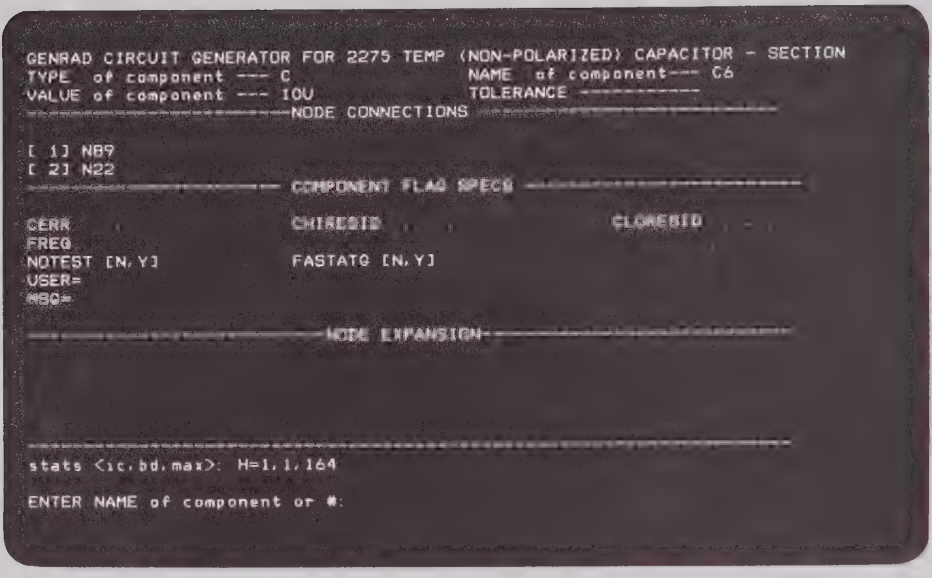
```
GENRAD CIRCUIT GENERATOR FOR 2275 TEMP EXTERNAL CONNECTION -- SECTION
TYPE  of component --- EXT              NAME of component --VCCP5

------------------------NODE CONNECTIONS--------------------------

[ 1] N21

----------------------------NODE EXPANSION------------------------



------------------------------------------------------------------
stats <ic,bd,max>: H=1,1,164

ENTER NAME of component or #
```

Figure 6-5. CKTGEN External Connection Display.



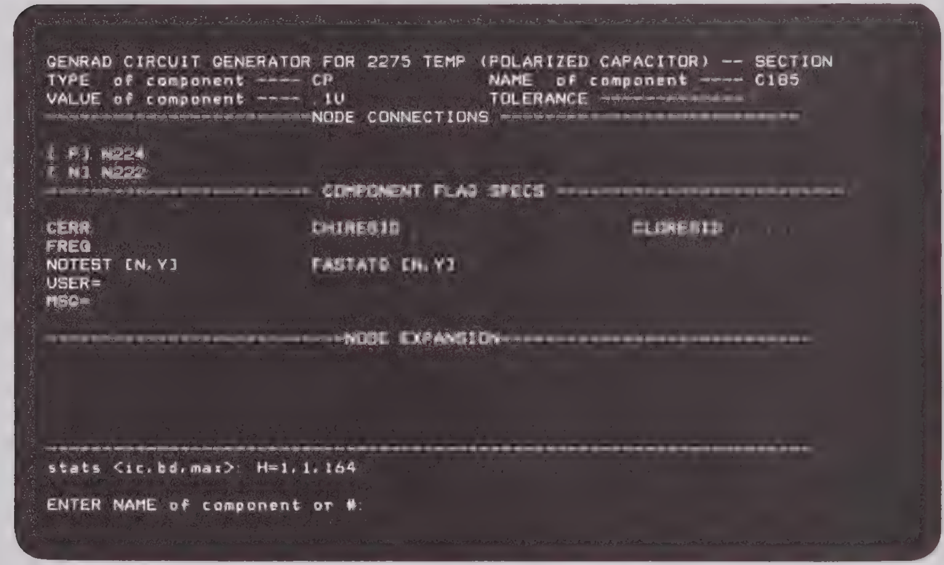
```
GENRAD CIRCUIT GENERATOR FOR 2275 TEMP RESISTOR -- SECTION
TYPE  of component ---- R               NAME  of Component --- R1
VALUE of component ---- 1K              TOLERANCE ------------
------------------------NODE CONNECTIONS -------------------------

[ 1] N15
[ 2] N16
------------------------ COMPONENT FLAG SPECS --------------------

RERR                     RHIRESID                  RLORESID
ILIM                     RLOV                      RHIV
NOTEST [N,Y]             FASTATG [N,Y]
USER=
MSG=

---------------------------NODE EXPANSION------------------------



-----------------------------------------------------------------
stats <ic,bd,max>: H=1,1,164

ENTER NAME of component or #
```

Figure 6-6. CKTGEN Resistor Display.



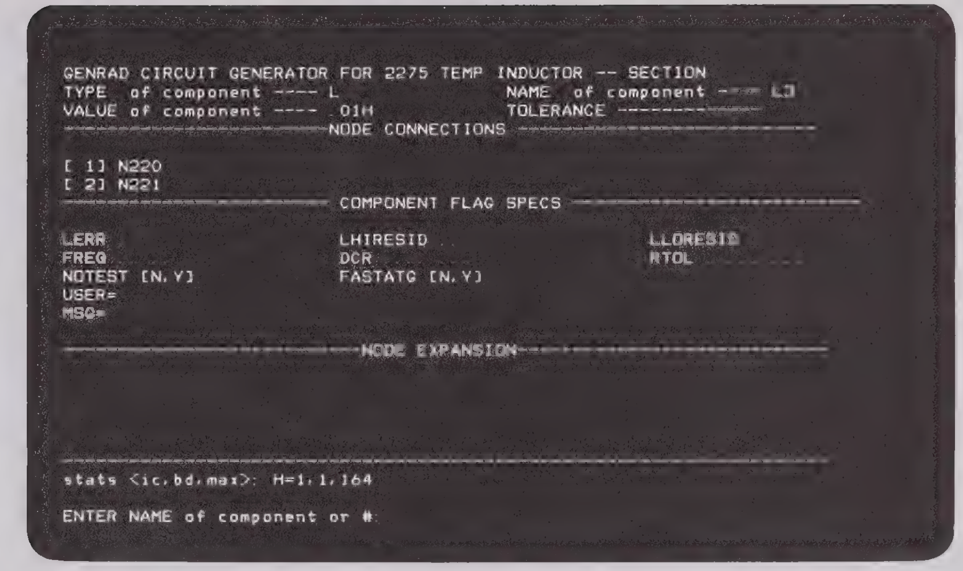
```
GENRAD CIRCUIT GENERATOR FOR 2275 TEMP (NON-POLARIZED) CAPACITOR - SECTION
TYPE  of component --- C                 NAME  of component--- C6
VALUE of component --- 10U               TOLERANCE -----------
-----------------------NODE CONNECTIONS --------------------------

[ 1] N89
[ 2] N22
----------------------- COMPONENT FLAG SPECS --------------------------

CERR                    CHIRESID                     CLORESID
FREQ
NOTEST [N,Y]            FASTATG [N,Y]
USER=
MSG=

--------------------------NODE EXPANSION--------------------------




---------------------------------------------------------------------
stats <ic.bd.max>: H=1,1,164

ENTER NAME of component or #:
```

Figure 6-7. CKTGEN (Non-Polarized) Capacitor Display.



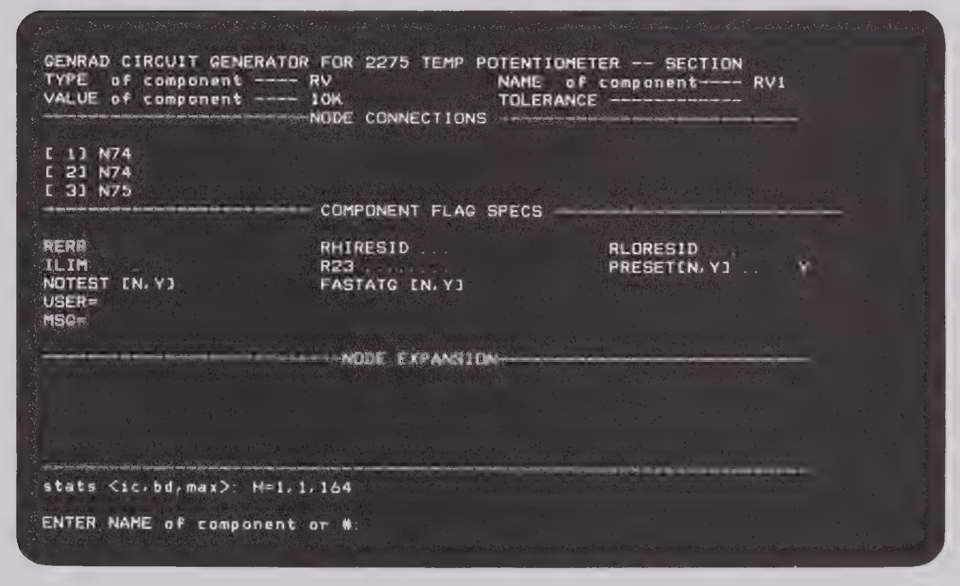
```
GENRAD CIRCUIT GENERATOR FOR 2275 TEMP (POLARIZED CAPACITOR) -- SECTION
TYPE  of component ---- CP               NAME  of component ---- C185
VALUE of component ----  1U              TOLERANCE -------------
-----------------------NODE CONNECTIONS --------------------------

[ P] N224
[ N] N222
----------------------- COMPONENT FLAG SPECS --------------------------

CERR                    CHIRESID                     CLORESID
FREQ
NOTEST [N,Y]            FASTATG [N,Y]
USER=
MSG=

--------------------------NODE EXPANSION--------------------------




---------------------------------------------------------------------
stats <ic.bd.max>: H=1,1,164

ENTER NAME of component or #:
```

Figure 6-8. CKTGEN (Polarized) Capacitor Display.



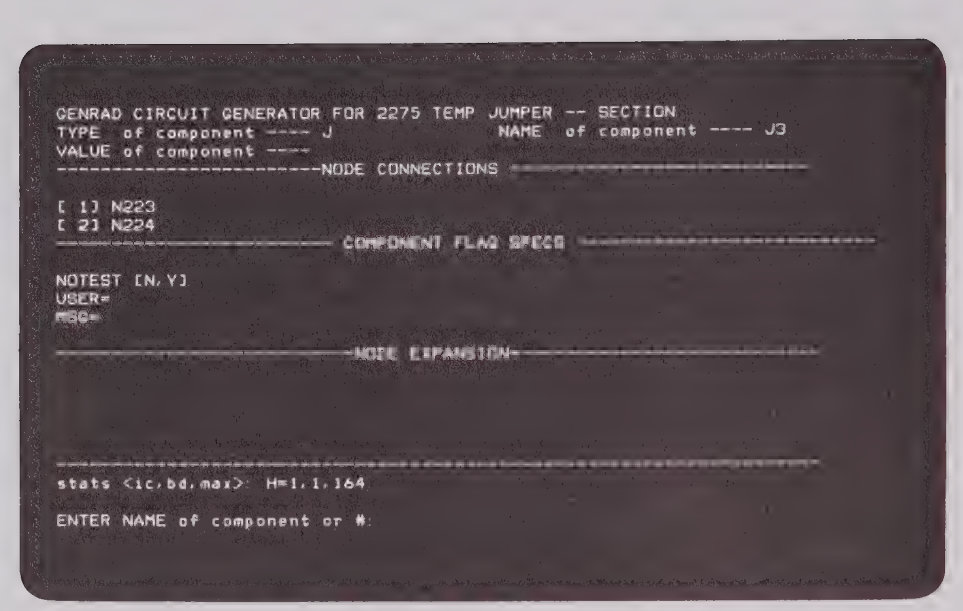
```
GENRAD CIRCUIT GENERATOR FOR 2275 TEMP INDUCTOR -- SECTION
TYPE  of component ---- L                 NAME  of component ---- L3
VALUE of component ---- .01H              TOLERANCE -----------
------------------------NODE CONNECTIONS ---------------------------

[ 1] N220
[ 2] N221
------------------------ COMPONENT FLAG SPECS -------------------------

LERR                     LHIRESID .                  LLORESID
FREQ                     DCR                         RTOL . . . . .
NOTEST [N,Y]             FASTATG [N,Y]
USER=
MSG=

--------------------------NODE EXPANSION------------------------------


---------------------------------------------------------------------
stats <ic,bd,max>: H=1,1,164

ENTER NAME of component or #:
```

Figure 6-9. CKTGEN Inductor Display.



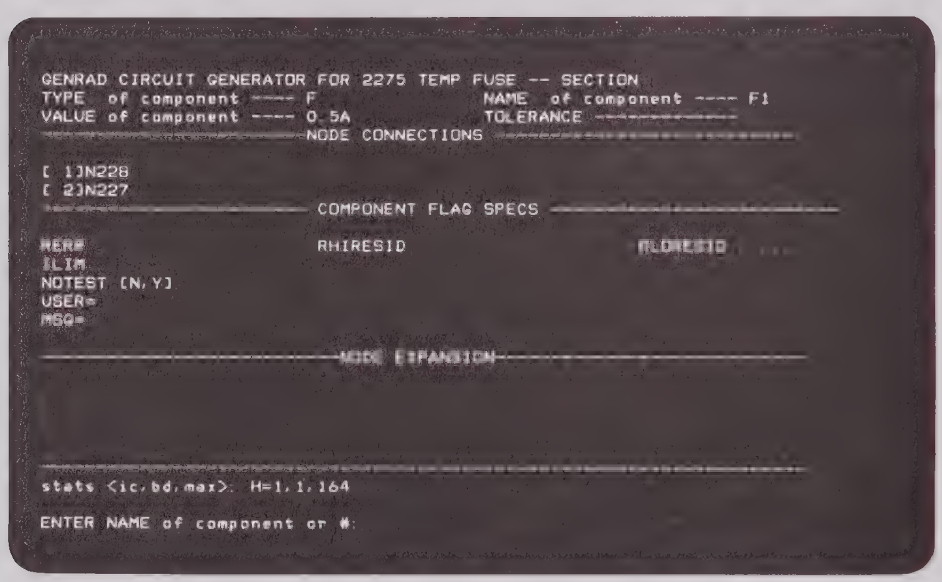
```
GENRAD CIRCUIT GENERATOR FOR 2275 TEMP POTENTIOMETER -- SECTION
TYPE  of component ---- RV                NAME  of component---- RV1
VALUE of component ---- 10K               TOLERANCE ------------
------------------------NODE CONNECTIONS ---------------------------

[ 1] N74
[ 2] N74
[ 3] N75
------------------------ COMPONENT FLAG SPECS -------------------------

RERR                     RHIRESID ...                RLORESID
ILIM                     R23 . . . . . .             PRESET[N,Y] ..    Y
NOTEST [N,Y]             FASTATG [N,Y]
USER=
MSG=

--------------------------NODE EXPANSION------------------------------


---------------------------------------------------------------------
stats <ic,bd,max>: H=1,1,164

ENTER NAME of component or #:
```

Figure 6-10. CKTGEN Potentiometer Display.



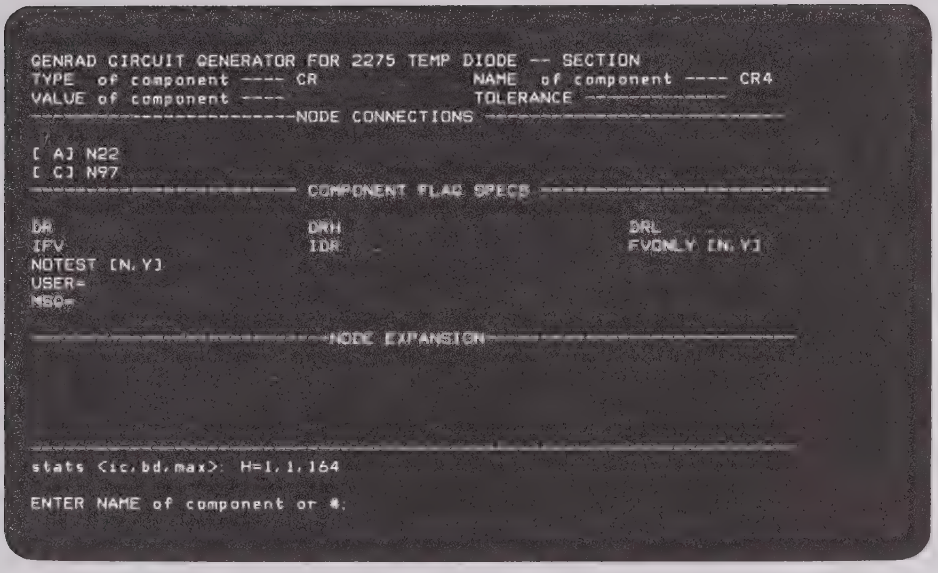
```
GENRAD CIRCUIT GENERATOR FOR 2275 TEMP JUMPER -- SECTION
TYPE  of component ---- J                NAME  of component ---- J3
VALUE of component ----
------------------------NODE CONNECTIONS ---------------------------

[ 1] N223
[ 2] N224
------------------------- COMPONENT FLAG SPECS ---------------------------

NOTEST [N,Y]
USER=
MSG=

--------------------------NODE EXPANSION--------------------------



---------------------------------------------------------------------
stats <ic,bd,max>: H=1,1,164

ENTER NAME of component or #:
```

Figure 6-11. CKTGEN Jumper Display.



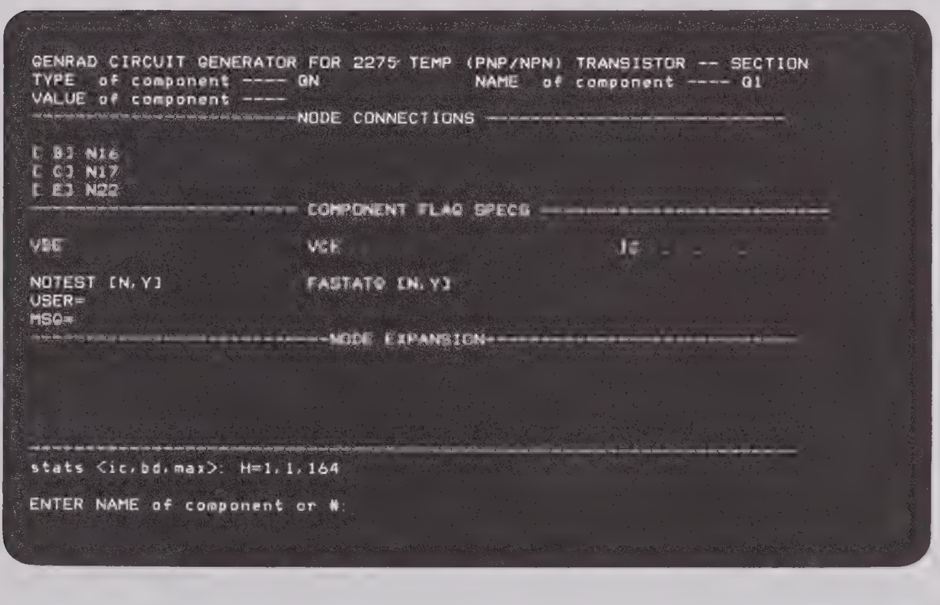
```
GENRAD CIRCUIT GENERATOR FOR 2275 TEMP FUSE -- SECTION
TYPE  of component ---- F                NAME  of component ---- F1
VALUE of component ---- 0.5A             TOLERANCE -------------
------------------------NODE CONNECTIONS ---------------------------

[ 1]N228
[ 2]N227
------------------------ COMPONENT FLAG SPECS ---------------------------

RERR                     RHIRESID                      RLORESID  ...
ILIM
NOTEST [N,Y]
USER=
MSG=

--------------------------NODE EXPANSION--------------------------



---------------------------------------------------------------------
stats <ic,bd,max>: H=1,1,164

ENTER NAME of component or #:
```

Figure 6-12. CKTGEN Fuse Display.



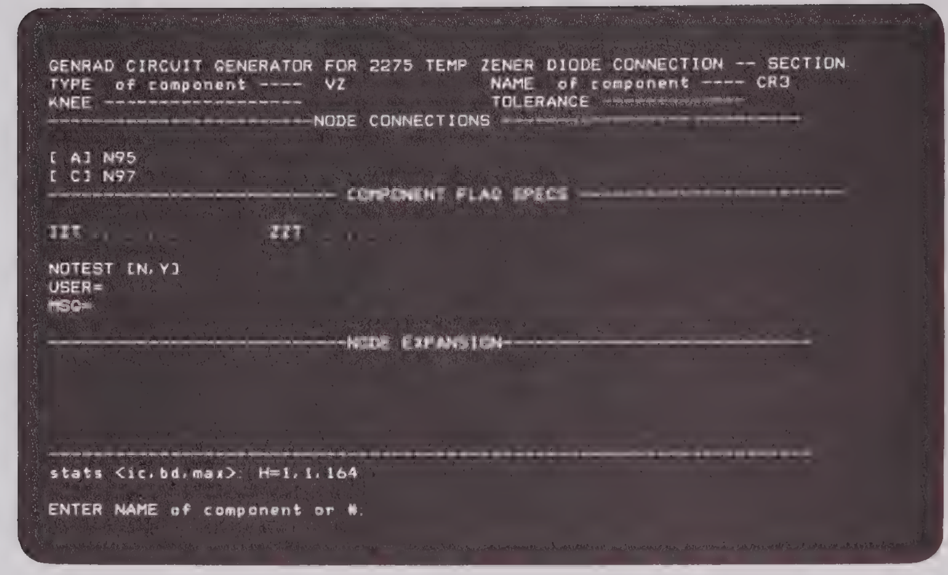
```
GENRAD CIRCUIT GENERATOR FOR 2275 TEMP DIODE -- SECTION
TYPE  of component ---- CR               NAME  of component ---- CR4
VALUE of component ----                  TOLERANCE ------------
--------------------------NODE CONNECTIONS ---------------------------

[ A] N22
[ C] N97
-------------------------- COMPONENT FLAG SPECS --------------------------

DR                        DRH                              DRL
IFV                       IDR                              FVONLY [N,Y]
NOTEST [N,Y]
USER=
MSG=

----------------------------NODE EXPANSION----------------------------




----------------------------------------------------------------------
stats <ic,bd,max>: H=1,1,164

ENTER NAME of component or #:
```

Figure 6-13. CKTGEN Diode Display.



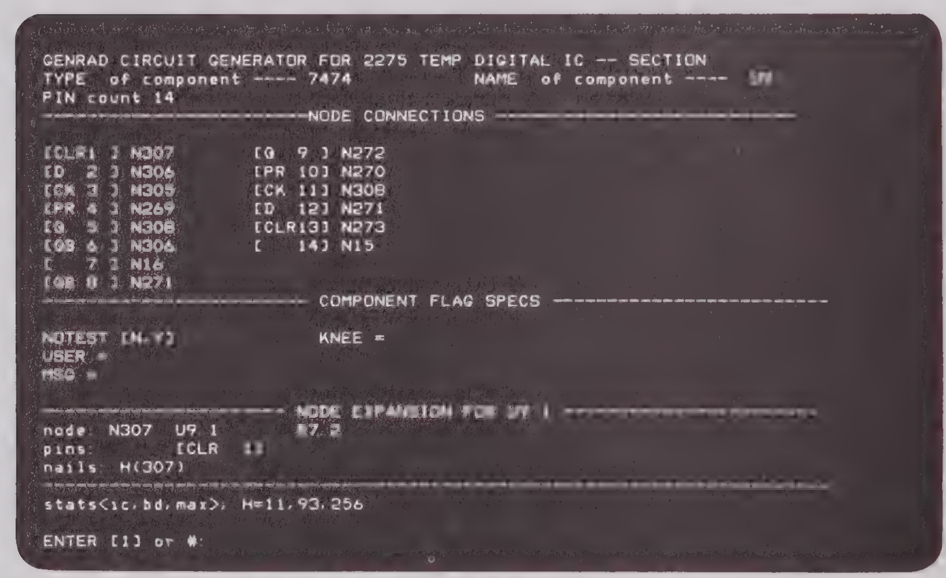
```
GENRAD CIRCUIT GENERATOR FOR 2275 TEMP (PNP/NPN) TRANSISTOR -- SECTION
TYPE  of component ---- QN               NAME  of component ---- Q1
VALUE of component ----
--------------------------NODE CONNECTIONS ---------------------------

[ B] N16
[ C] N17
[ E] N22
-------------------------- COMPONENT FLAG SPECS --------------------------

VBE                       VCE                              IC

NOTEST [N,Y]              FASTATG [N,Y]
USER=
MSG=
----------------------------NODE EXPANSION----------------------------




----------------------------------------------------------------------
stats <ic,bd,max>: H=1,1,164

ENTER NAME of component or #:
```

Figure 6-14. CKTGEN Transistor Display.



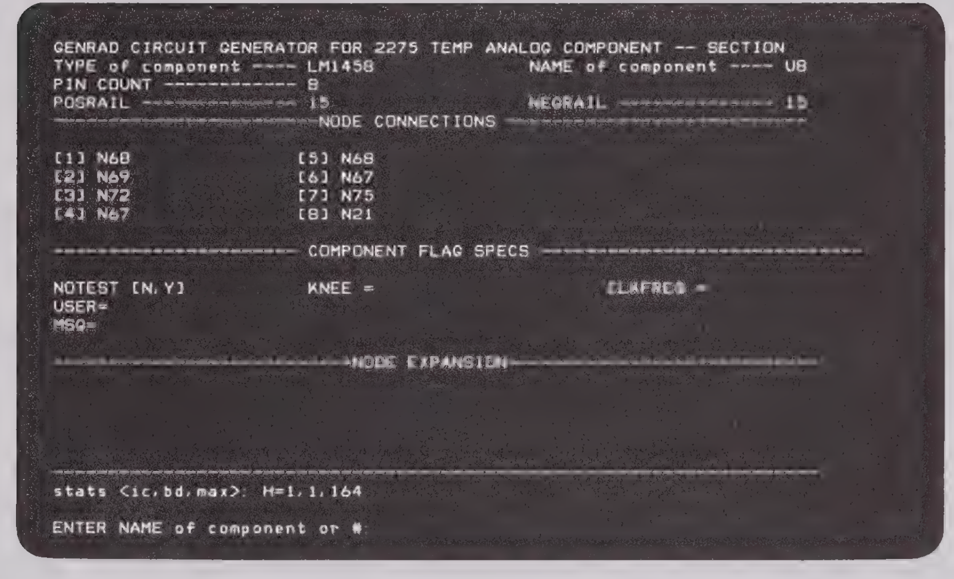
```
GENRAD CIRCUIT GENERATOR FOR 2275 TEMP ZENER DIODE CONNECTION -- SECTION
TYPE  of component ----  VZ                 NAME  of component ---- CR3
KNEE -----------------                      TOLERANCE ----------
-----------------------NODE CONNECTIONS ---------------------------

[ A] N95
[ C] N97
------------------------- COMPONENT FLAG SPECS ----------------------

IZT . . . . .          ZZT . . . .

NOTEST [N,Y]
USER=
MSG=

-------------------------NODE EXPANSION----------------------------




-------------------------------------------------------------------
stats <ic,bd,max>: H=1,1,164

ENTER NAME of component or #.
```

Figure 6-15. CKTGEN Zener Diode Display.



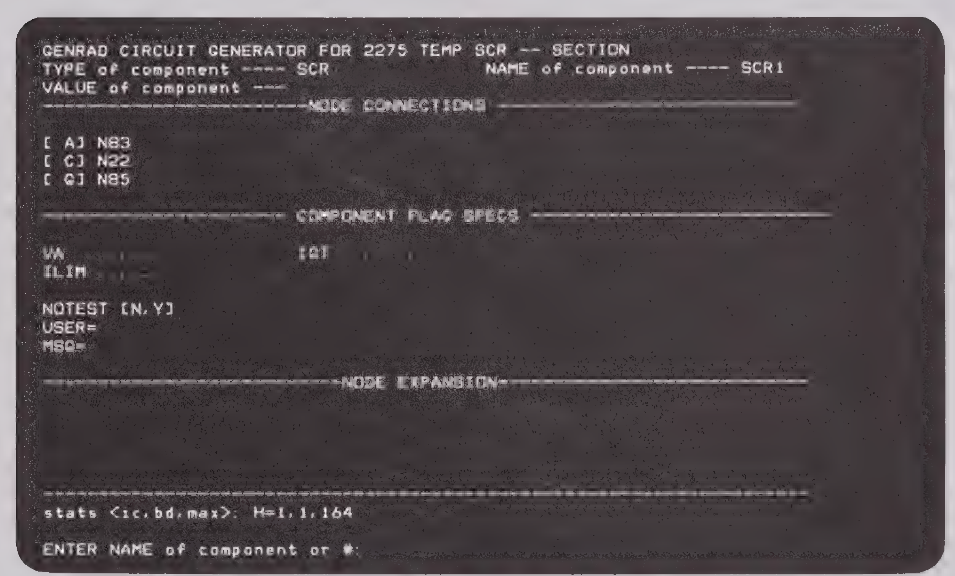
```
GENRAD CIRCUIT GENERATOR FOR 2275 TEMP DIGITAL IC -- SECTION
TYPE  of component ---- 7474            NAME  of component ----  U9
PIN count 14
-----------------------NODE CONNECTIONS ---------------------------

[CLR1  ] N307        [Q   9 ] N272
[D   2 ] N306        [PR  10] N270
[CK  3 ] N305        [CK  11] N308
[PR  4 ] N269        [D   12] N271
[Q   5 ] N308        [CLR13] N273
[QB  6 ] N306        [    14] N15
[    7 ] N16
[QB  8 ] N271
----------------------- COMPONENT FLAG SPECS ------------------------

NOTEST [N,Y]               KNEE =
USER =
MSG =

---------------------- NODE EXPANSION FOR U9 1 ----------------------
node: N307  U9 1       U7 2
pins:        [CLR  1]
nails: H(307)
--------------------------------------------------------------------
stats<ic,bd,max>; H=11,93,256

ENTER [1] or #:
```

Figure 6-16. CKTGEN Digital IC Display.



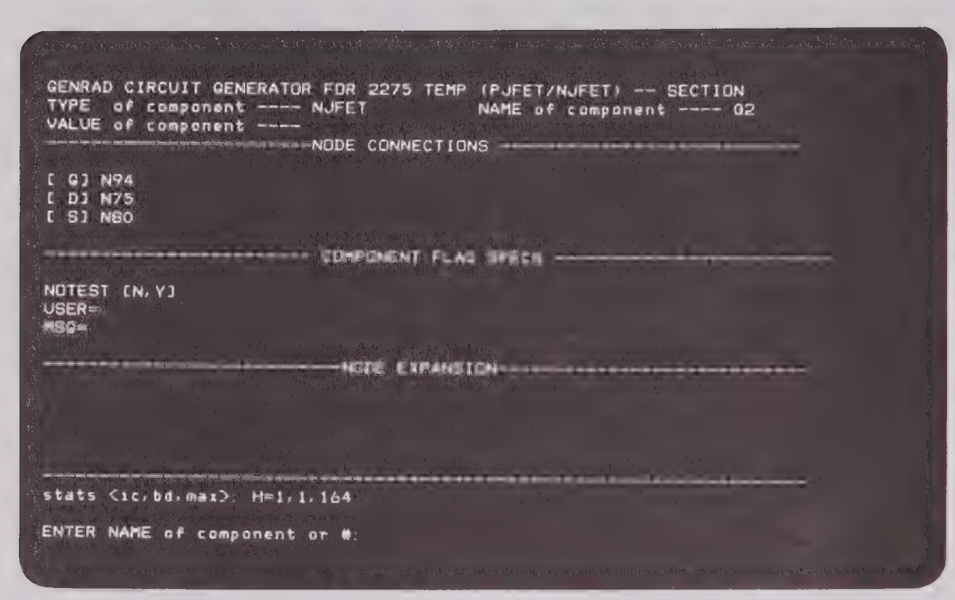
```
GENRAD CIRCUIT GENERATOR FOR 2275 TEMP ANALOG COMPONENT -- SECTION
TYPE of component ---- LM1458            NAME of component ---- U8
PIN COUNT ------------ 8
POSRAIL -------------- 15                NEGRAIL -------------- 15
----------------------NODE CONNECTIONS ---------------------------

[1] N68                [5] N68
[2] N69                [6] N67
[3] N72                [7] N75
[4] N67                [8] N21

--------------------- COMPONENT FLAG SPECS ---------------------------

NOTEST [N,Y]           KNEE =                    CLAFREQ =
USER=
MSG=

---------------------------NODE EXPANSION-------------------------



-------------------------------------------------------------------
stats <ic,bd,max>: H=1,1,164

ENTER NAME of component or #:
```

Figure 6-17. CKTGEN Analog Component Display.



```
GENRAD CIRCUIT GENERATOR FOR 2275 TEMP SCR -- SECTION
TYPE of component ---- SCR              NAME of component ---- SCR1
VALUE of component ---
-----------------------NODE CONNECTIONS --------------------------

[ A] N83
[ C] N22
[ G] N85

--------------------- COMPONENT FLAG SPECS --------------------------

VH                     IGT
ILIM

NOTEST [N,Y]
USER=
MSG=

---------------------------NODE EXPANSION-------------------------



-------------------------------------------------------------------
stats <ic,bd,max>: H=1,1,164

ENTER NAME of component or #:
```

Figure 6-18. CKTGEN SCR Display.



```
GENRAD CIRCUIT GENERATOR FOR 2275 TEMP (PJFET/NJFET) -- SECTION
TYPE  of component ---- NJFET            NAME of component ---- Q2
VALUE of component ----
----------------------NODE CONNECTIONS ----------------------

[ G] N94
[ D] N75
[ S] N80

---------------------- COMPONENT FLAG SPECS ----------------------

NOTEST [N,Y]
USER=
MSG=

------------------------NODE EXPANSION------------------------




---------------------------------------------------------------
stats <ic,bd,max>: H=1,1,164

ENTER NAME of component or #:
```

Figure 6-19. CKTGEN NJFET/PJFET Display.

6.6.5 Report Page

A file (.CRP) containing statistics can be enabled to provide information to aid you in making nail allocation decisions when the .CKT file is created. The .CRP file is enabled on the CKTGEN monitor page. This file is displayed when the command REPORT is issued from anywhere in CKTGEN as long as it is not entered as a response to a MSG= or USER= flagspec. For a 2270 REPORT, nail types consist of A (analog), D (digital), and H (hybrid). For the other testers, the nail types can include CD (clock driver), CS (clock sensor), T (trigger) and H (hybrid). The report consists of a list and totals of ICs, nails, nodes, etc in a specific category. Each list is presented in alphabetical order. Empty lists are displayed with the word "NO" rather tnan "TOTAL OF n".

In the example REPORT listing that follows, the report is shown on the left, and any statistic that is not self explanatory is described on the right.

```
GENRAD CKTGEN REPORT FOR 2272 TEMP FOR boardname

STATS <allocated, maximum>:
CD = 10,16   CS = 8,8   T = 14,32   F = 200,384   H = 1023,3456
```

The first specification is the number of a given nail type that has been allocated on this board; the second number is the capacity for this particular tester and is known by the system software.

```
TOTAL OF 3 ICs ALLOCATED FAST NAILS (* indicates FAST test only):
U21 (M6800), U45 (I8085*), U75 (MC6802)
```

The above ICs have pins that have been allocated Fast nails, regardless of whether or not a fast test is going to be selected.

```
TOTAL OF 2 ICs ALLOCATED INSUFFICIENT NAILS FOR TESTING:
U20 (I8085), U79 (MC6802)
```

Based on the known library tests for the above ICs, certain nail types are required, but have not been allocated for one or more reasons. Such reasons include among others: no more of a nail type are available for allocation; a pin on this IC requires a nail of greater capability; etc.

```
TOTAL OF 2 IC TYPES HAVING CKTGEN TEST SKELETONS IN dtlname.DTL:
I8049 (U10, U22, U32), I8080 (U16, U17)
```

The above ICs have test skeletons that have been INSERTED in the user digital library for CKTGEN purposes. ATG tests, however, need to be created for these ICs before test program preparation is complete.

```
TOTAL OF 2 IC TYPES HAVING NO CKTGEN TEST SKELETONS IN ANY DTL:
I8047 (U90, U92, U93), I8380 (U86, U87)
```

The above ICs have not yet been DEFINEd and INSERTed.

```
TOTAL OF 1 NODES HAVING NO PINS:
FOO=(H62), (F)33, H(22);
```

Any nodes listed here have nails allocated as shown, but no component pins. This is because the component(s) have been REMOVEd.

If you request a .CRP file (from within the CKTGEN monitor page), the report for the final state of the .CIR file is written to the .CRP file on exit from CKTGEN. The file format is as follows:

```
GENRAD CKTGEN REPORT FOR 2272 TEMP FOR boardname

          10 CD nails out of   16 CD nails have been allocated.
           8 CS nails out of    8 CS nails have been allocated.
          14  T nails out of   32  T nails have been allocated.
         200  F nails out of  384  F nails have been allocated.
        1023  H nails out of 3456  H nails have been allocated.
```

```
TOTAL OF 3 ICs ALLOCATED FAST NAILS

IC NAME  IC TYPE FAST TEST ONLY

U21      MC68000
U45      I8085        *
U75      MC6802


TOTAL OF 2 ICs ALLOCATED INSUFFICIENT NAILS FOR TESTING

IC NAME  IC TYPE

U20      I8085
U79      MC6802


TOTAL OF 2 IC TYPES HAVING CKTGEN TEST SKELETONS IN dtlname.DTL

IC NAME  IC TYPE

I8049    U10, U22, U32
I8080    U16, U17


TOTAL OF 2 IC TYPES HAVING NO CKTGEN TEST SKELETONS IN ANY DTL


IC NAME  IC TYPE

I8047    U90, U92, U93
I8380    U86, U87


TOTAL OF 1 NODES HAVING NO PINS

IC NAME  IC TYPE

FOO      (H62), (F)33, H(22)
```

6.7 DETAILED CKTGEN

6.7.1 General

The capabilities of CKTGEN go beyond the basic features described for the new user in paragraph 6.5. With the flexible command set (paragraph 6.8) and node expansion capabilities, you can skip (GOTO) to any part of the circuit and examine the entire node connected to that point, or look at the connections for a given component. You can then modify the connections if necessary, and even CAP the node to prevent CKTGEN from adding or deleting nails to/from it.

The circuit-description characteristics of a component or a node and any associated nails can be interrogated (SHOW command) and/or changed using nail allocation commands (DENOTE, ADD, REMOVE, CAP, etc). A node name is the connection name on an IC pin. Every circuit node has a node name, either provided by the user or created by the .CKT File Reader. Component pins for which the user has entered no connection information have no associated nodes. Among some of the node expansion and nail capabilities are:

a node can be displayed and a component pin can be ADDed or DELeted.

a node can be CAPped (no other nails can be added or removed by CKTGEN).

the aspects of nails (refer to the DENOTE command) can be changed.

These commands are listed and described in paragraph 6.8

CKTGEN includes a report page (para 6.6.5) to aid the user in making nail allocation decisions. This page displays:

nail counts allocated by type,

ICs with fast nails,

ICs with fast tests but insufficient fast nails,

IC types with CKTGEN test skeletons,

nodes with no component pins.

CKTGEN interacts with the system, update, and digital user libraries. This capability allows you to DEFINE an IC not currently known to the libraries. CKTGEN makes use of the information available in the HEAD sections of the digital libraries. For digital ICs not found in any library, CKTGEN provides for insertion of a CKTGEN "test skeleton" into the user DTL (para 6.7.4). The test skeleton consists of pin-connection and other component information and allows the circuit description to be completed (a .CKT file made) even if ATG tests have not yet been generated for the new IC types.

CKTGEN will reference the digital libraries to search for information about an IC. The user digital library is specified on the CKTGEN monitor page. The system and update libraries are specified on the SETUP monitor page. The user library will be used to store CKTGEN test skeletons. CKTGEN can be used to modify a test skeleton, as long as it was created and placed in the user library with CKTGEN. All other tests are "read only".

6.7.2 CKTGEN Nail Allocation and Aspects

CKTGEN normally allocates a nail to every node when the node is created. There are 2 aspects to a nail, its number and its type. The types include:

CD	clock driver
CS	clock sensor
T	trigger
F	fast
H	hybrid
D	digital
A	analog

CKTGEN will assign both aspects, resulting in a number that is unlikely to change and a type that satisfies the testing requirements. For 2271 and 2272 testers, CKTGEN will attempt to update nail types without modifying nail numbers. With a 2270, CKTGEN attempts to match nail numbers to nodes based on the perceived testing requirements.

All nail types and numbers in CKTGEN are in one of 2 states, either tentative or unchangeable as follows:

Tentative	CKTGEN is allowed to change the aspects, but CKTGEN will attempt to leave the current assignment untouched.
Unchangeable	CKTGEN cannot change the aspects.

It is possible to change nail numbers and nail types independently (refer to the DENOTE command). Both type and number are tentatively assigned when CKTGEN automatically allocates a nail.

CKTGEN uses an algorithm for picking nail numbers that tends to minimize the need for further number changes during the entire run of CKTGEN. Tentative nail numbers and nail types are displayed within parentheses and can be changed by CKTGEN; otherwise the number or type is considered unchangeable. The possible combinations are as follows:

Representation	Type	Number
(H24)	Tentative	Tentative
(H)24	Tentative	Unchangeable
H(24)	Unchangeable	Tentative
H24	Unchangeable	Unchangeable

CKTGEN automatically allocates a nail number whenever it creates a new node using the following algorithm:

1. If the node name consists of a string with no numeric digits then CKTGEN will assign the highest unused overflow nail number for that node. The overflow numbers run from 6000 down. For example: the fourth overflow nail is probably assigned to (5997).

2. If the node name contains digits (where dddd are the rightmost digits that are less than 6000) then CKTGEN assigns the nail number dddd to that node, unless nail number dddd was being used by a node whose name contains digits (including Ndddd), or the user had manually made the nail number dddd unchangeable. If the node did not receive the number dddd, the node would be assigned the largest unused overflow nail number.

3. If the node name consists of N<dddd> then CKTGEN assigns the nail number dddd to that node and marks both it and its type as tentative. If nail dddd was already hand allocated and unchangeable, then node N<dddd> would be assigned the largest unused overflow nail number. If nail dddd was already allocated, but not marked unchangeable, then the old dddd nail would have its number part changed to the largest available overflow nail number.

CKTGEN assigns nail types to nodes based on the pins currently connected to the nodes. If there are no nails of a given type available CKTGEN will use the best available nail (or none, if none exist) and mark that node as lacking capabilities. Whenever a new pin (including the first one) is connected to a node, the tentative nail type may be upgraded to satisfy the new testing requirements.

CKTGEN will reference the DTL library entries to decide whether FAST or DIGITAL nail capabilities are required on a node. If there are analog components other than resistors on a node with digital ICs, CKTGEN will add ANALOG capabilities to the node. If there are only analog components on a node, CKTGEN will assign a nail with analog capability to the node. For a 2270 system, CKTGEN attempts to use the nail allocation defined by the N<ddd> nail numbers allocated by the user. (CKTGEN attempts to leave a hybrid nail on a pure digital node if ddd is in the hybrid range.) CKTGEN will add clock and trigger nails to a node in increasing order using their real numbers.

Nails read from temporary %ADAPTOR sections are all tentative when they are read. Nails read from target or 2270 %ADAPTOR sections are all unchangeable. As mentioned earlier, it is also possible to make nail numbers tentative or unchangeable.

6.7.3 CKTGEN and the Digital Libraries

When the node connections for a component are to be entered and that IC is known to either the system, update or user library, CKTGEN will display pin information from the DTL in the NODE CONNECTIONS area of the screen (see Figure 6-20). Pin information consists of the following:

a pin count,
FREQuency - (clock-input testing rate),
normal test status for the IC,
a pin name and number for each IC pin.

The FREQ message will be displayed only if the information exists in the DTL. Library information for each pin will be displayed inside the brackets associated with the pin. As you respond to the prompt for a node specification at the bottom of the screen, expanded node information is displayed in the lower section of the screen.

In the following example an IC named U21 (TYPE = MC68000) will be placed in the CKTGEN database. (The IC is known to the digital library. The node connections are now added.)

Header Information

The library header for this example indicates the following:

the component type - (MC68000 for this example)

the target system type (from the SETUP page) - 2272 here,

a pin count of 64 - (the library knew this and set the format)

a testing clock rate of 2.0 MEG - (the library knew this)

the component name is U21 - (specified as a response to the NAME = prompt)

that a "NORMAL TEST EXISTS" - (other messages in this position include "FAST TEST ONLY", and CKTGEN TEST SKELETON"). The library supplies this information.

Node Connections

Figure 6-20 shows that node connections for several of the pins have already been entered. Each entry can consist of the following fields and is described in more detail in the next paragraph:

pin name (truncated to the first 3 letters)
pin number
a special symbol
a node number or node name

The node names are alphanumeric names of up to 8 characters (e.g. N29, FOOBAR) or pin numbers of a component (e.g. U14.1) whose node name has been previously defined. The connection will be displayed by its node name in the NODE CONNECTIONS area of the screen. The prompt for each pin is located at the bottom of the screen. The current prompt is for pin 2. This pin is displayed in intensified video (simulated by dashes here).

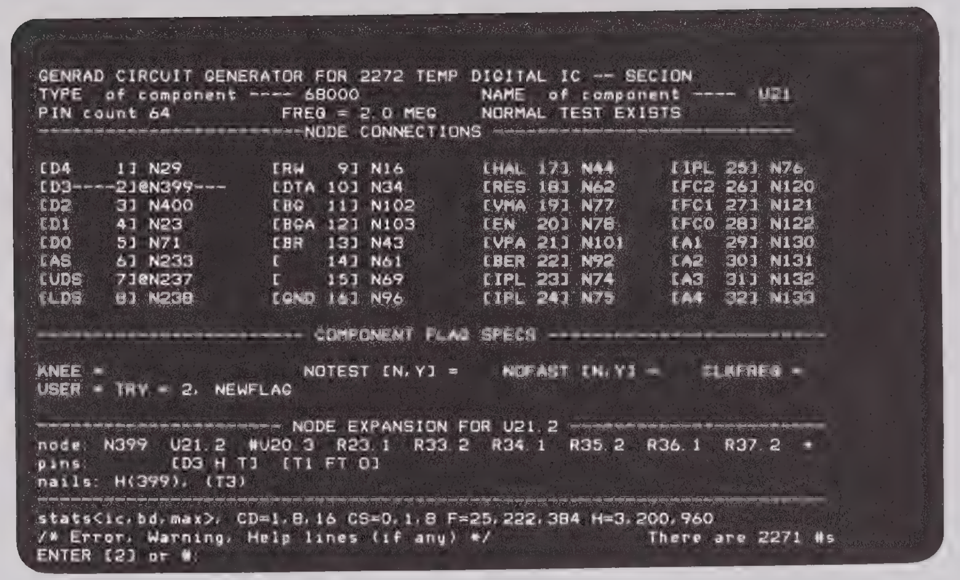
A node may be prefaced by a special symbol. The symbols are listed and described following the Node Expansion description below because most of the symbols apply to that display.

Component Flagspecs.

The flagspecs shown in Figure 6-20 are described in the CDL description (paragraph 6.12.6).

Node Expansion

The NODE EXPANSION display section of the screen shows the status of the connections of the node associated with the named pin (NODE EXPANSION FOR U21.2 in this example). This section of the display consists of 3 classes of information. These are: node, pins, and nails.



```
GENRAD CIRCUIT GENERATOR FOR 2272 TEMP DIGITAL IC -- SECION
TYPE  of component ---- 68000          NAME  of component ----  U21
PIN count 64          FREQ = 2.0 MEG    NORMAL TEST EXISTS
------------------------NODE CONNECTIONS -------------------------

[D4    1] N29         [RW    9] N16        [HAL 17] N44      [IPL 25] N76
[D3----2]@N399---     [DTA 10] N34         [RES 18] N62      [FC2 26] N120
[D2    3] N400        [BG  11] N102        [VMA 19] N77      [FC1 27] N121
[D1    4] N23         [BGA 12] N103        [EN  20] N78      [FC0 28] N122
[D0    5] N71         [BR  13] N43         [VPA 21] N101     [A1  29] N130
[AS    6] N233        [    14] N61         [BER 22] N92      [A2  30] N131
[UDS   7]@N237        [    15] N69         [IPL 23] N74      [A3  31] N132
[LDS   8] N238        [GND 16] N96         [IPL 24] N75      [A4  32] N133

----------------------- COMPONENT FLAG SPECS ------------------------

KNEE =                 NOTEST [N,Y] =    NOFAST [N,Y] =    CLKFREQ =
USER = TRY = 2, NEWFLAG

--------------------- NODE EXPANSION FOR U21.2 ----------------------
node: N399  U21.2  #U20.3  R23.1  R33.2  R34.1  R35.2  R36.1  R37.2  +
pins:       [D3 H T]  [T1 FT 0]
nails: H(399), (T3)
---------------------------------------------------------------------
stats<ic,bd,max>, CD=1,8,16 CS=0,1,8 F=25,222,384 H=3,200,960
/* Error, Warning, Help lines (if any) */              There are 2271 #s
ENTER [2] or #:
```

Figure 6-20. Adding An IC To The CKTGEN Database.

node: The node name (N399) is listed first and is followed by a line of components connected to the node. A + symbol indicates a line overflow. Only one line of components can be displayed at a time. Issue a #MORE command to view any additional components tied to this node. Digital IC connections are displayed before analog component connections. A #NOEXPAND command can be entered to suppress this data and reduce operating time if you are not interested in it. (Refer to the #EXPAND and #NOEXPAND commands in the command summary.)

In this example, one of the pins (U20.3) is preceded by a # symbol. As described below, this symbol indicates that a nail allocated to this node does not have the capability to be used in the preferred test for IC 20. Note that this information is displayed even though U20 is not the currently displayed component.

If no node has yet been entered for the pin, the display will show only pin information for the IC.

pins: pinname, type, and nail type requirements for each component on the node having a DTL entry. Pin information is displayed within brackets, directly under the corresponding pin number on the node line.

nails: nails assigned to the node. Parentheses indicate that the enclosed aspect(s) (nail number and/or nail type) are tentative and can be changed by CKTGEN. A + symbol indicates a line overflow. If the word CAPPED appears, it indicates the node is capped and no nails can be added or removed (refer to the CAP command).

There are several node expansion and node control commands that are applicable to the NODE EXPANSION capabilities.

Node-Display Special Symbols

The following special symbols may appear in the NODE CONNECTIONS or NODE EXPANSION area. These symbols apply to both analog and digital components.

# indicates a component pin on a node that probably has no nail that can be used in the preferred test for the component.

The "2271 problems" message is displayed if the 2271 target is active but not the currently displayed target and the nails assigned to the IC for the 2271 are not of sufficient capability to test the component (i.e. if # is displayed for some pin on the IC when TARGET= an active target not currently displayed).

@ indicates a node that contains a component pin that probably has no nail that can be used in the preferred test for the component (but the component is not the one currently displayed).

() around nail numbers and types indicates that the numbers and types are tentative and can be changed by CKTGEN. If the node is in the CAPped state, the nail line would appear: "nails: CAPPED H399, CD3".

+ at the end of the mode, pins or nails lines indicates a line overflow. Refer to the MORE command.

Statistics, Error, Warning Help Lines

The bottom section of the display contains miscellaneous information consisting of:

stats: number of nails allocated so far for each nail type, for the currently displayed ic and the board, and the number of each nail type available with the tester. In Figure 6-20 for example, the statistic CD=1,8 16 indicates that 1 clock driver nail has been allocated for this IC, 8 for the board, and that 16 are available for this tester. These statistics are printed in the .CRP file.

Miscellaneous Messages: Any error or warning messages are displayed on the line beneath the stats line. Also, if there are multiple targets, the tester types are noted on this line (as the 2271 is noted in this example). The response prompt appears on the bottom line of the display.

6.7.4 Defining An IC

When a new IC is to be inserted in the CKTGEN data base, CKTGEN will search the libraries (in the order: user, update, system) to determine if a test for this IC type exists. If no test is found, CKTGEN will prompt:

```
ENTER ANALOG OR DIGITAL (A/D) or #:
```

For an analog component, CKTGEN will display the analog component page and the ENTER NAME prompt. For a digital component, CKTGEN will display a blank DEFINE TYPE digital IC page and the ENTER NAME prompt. The DEFINE TYPE command is used to save type-specific information in the user DTL. You can then DEFINE and INSERT the circuit connections for this IC into the user library in the form of a test skeleton. (A user library file created previously using the DTL mode, must be available to contain the inserted test skeleton.)

Figure 6-21 shows the format of the DEFINE TYPE page. First, header information consisting of the component type, pin count and test frequency (if any) is entered. When the pin count is entered, the specified entry blocks for the specified number of pins appears in the PIN DEFINITIONS area of the screen. Then, the data for each pin is entered in the PIN DEFINITIONS section of the screen. When the page is completed, a CKTGEN test skeleton for this IC can be INSERTed into the user digital library from the DEFINE TYPE page. If the type definition is modified while this page is displayed, the message "MODIFIED" will be displayed on the right side of the PIN COUNT line. Also, if the the PIN COUNT definition is created while this page is displayed, the message "NEW" will be displayed on the right side of the PIN COUNT line.

The DEFINE TYPE command can also be used to examine an existing system or update library component. These components can not be modified. A message will be displayed on the right side of the TYPE= line stating: "READ ONLY". Since such a test can not be modified, CKTGEN will prompt only ENTER NAME or #:

Nail/Pin Definitions

The characteristics of each component pin are entered beginning with the first pin (see Figure 6-21). This information includes:

pin names if desired.

pin number

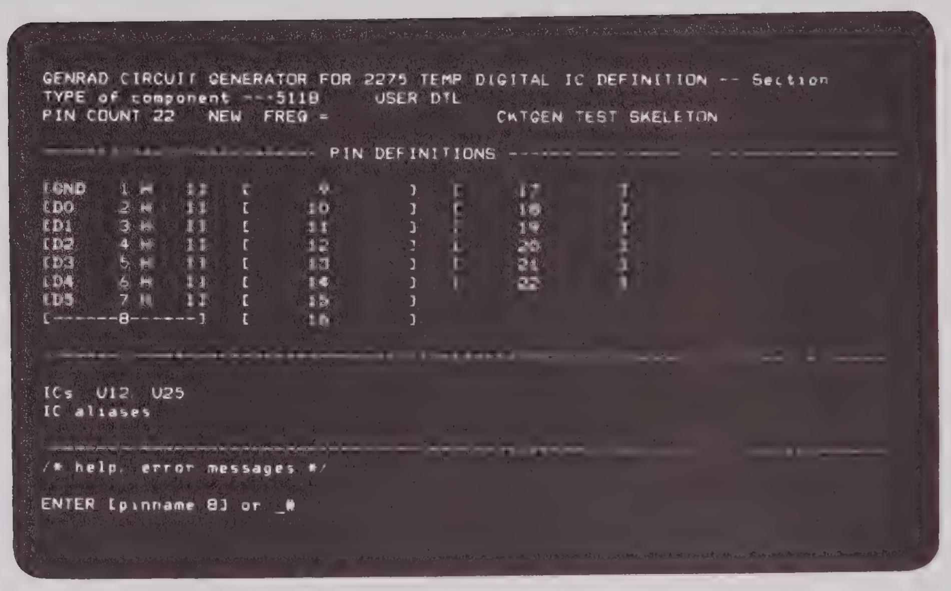
special and fast nail types required (CD, CS, T, F)
(hybrid, digital or analog pins do not need to be specified.)

pin type (I: for INPUT, O: for OUTPUT, T: 3-state, E: open emitter,
C: open collector)

Only the first of 32 pins can be displayed. For a component containing more pins, the display shows the remaining pins as soon as pin 33 (or higher) is prompted.

ICs: This line specifies other ICs of this type currently in the .CIR file, if any.

IC aliases: This line lists the other names that exist for this generic IC type.



```
GENRAD CIRCUIT GENERATOR FOR 2275 TEMP DIGITAL IC DEFINITION -- Section
TYPE of component ---511B      USER DTL
PIN COUNT 22   NEW  FREQ =                   CKTGEN TEST SKELETON

---------------------------- PIN DEFINITIONS ----------------------------

[GND    1 H   I]   [      9       ]  [     17      ]
[D0     2 H   I]   [     10       ]  [     18      ]
[D1     3 H   I]   [     11       ]  [     19      ]
[D2     4 H   I]   [     12       ]  [     20      ]
[D3     5 H   I]   [     13       ]  [     21      ]
[D4     6 H   I]   [     14       ]  [     22      ]
[D5     7 H   I]   [     15       ]
[-------8------]   [     16       ]

-------------------------------------------------------------------------

ICs  U12  U25
IC aliases

-------------------------------------------------------------------------
/* help, error messages */

ENTER [pinname 8] or _#
```

Figure 6-21. Defining an IC.

Defining a Pin - An Example

Refering to Figure 6-21, the user response to the prompt provided for pin 8 could be as follows:

```
ENTER [pinname 8] or #: D6
```

CKTGEN prompts for each entry one by one:

```
ENTER [8 CTFH] or #: H
```

```
ENTER [8 IOTCE] or #: I
```

For quicker entry, you can also respond to the first prompt:

```
ENTER [pinname 8] or #: D6,H,I
```

This assigns a hybrid nail type, pin type INPUT, and pin name DG. The information is filled in beside the corresponding pin number and then you are prompted for pin 9.

Once you have defined the IC and are satisfied with all of the entries, you can use the INSERT command to insert a new or modified definition into the user DTL. If you attempt to leave the page of a NEW or MODIFIED CKTGEN test skeleton, CKTGEN will write the warning that to not insert will result in the loss of the new skeleton or modification, and the prompt "INSERT this type definition [Y,N]:" will appear. If yes, CKTGEN will insert a CKTGEN test

The CKTGEN test skeleton consists of an IC HEADER section, and a dummy MAIN section. CKTGEN creates the proper format and does not call the digital syntax checker. The User DTL is marked as having this CKTGEN test skeleton inserted by CKTGEN. A CKTGEN test skeleton can be replaced by CKTGEN only under the following conditions:

the CKTGEN test skeleton can only be inserted into the User library

only a CKTGEN test skeleton inserted by CKTGEN can be replaced by CKTGEN.

The CKTGEN test skeleton written for an MC6800 would be:

```
/* CKTGEN TEST SKELETON FOR TYPE MC6800 */
.HEAD;
        .SKELETON; /* REMOVE SKELETON AFTER COMPLETING TEST BY ADDING
          INHIBITS AND  DISABLES */
        .SIZE 40;
        .FTEST (F=2 3 5-7 9 25, C=2 9 25, T=2 9 22-25);
        .INPUT (2=HLT,3=C1,4=IRQ,6=NMI,8=RAS);
        .OUTPUT (5=VMA,7=BA);
.END HEAD;
.MAIN;
.END MAIN;
```

Note that this CKTGEN test skeleton will consist of a HEAD section specifying pin and nail requirements, and an empty MAIN section. These CKTGEN test skeletons will not contain INHibit or DISABle sections, and as mentioned earlier, will be marked as CKTGEN test skeletons in the header section and in

6.8 CKTGEN COMMANDS SUMMARY

6.8.1 General

CKTGEN has several commands that can be used to perform various functions. These commands are grouped and described by function in the following paragraphs.

NOTE

The following commands can only be executed by entering the command mode. The command mode is entered by typing either a <CTRL G> or a <#> symbol. Read paragraph 6.9 before using any of these commands.

6.8.2 Node Expansion Control Commands

The following commands are valid on all CKTGEN component pages, and are associated with the NODE EXPANSION section of a component page display.

EXPAND and NOEXPAND: The NODE EXPANSION section of the display shows the device pin connections and temporary nails attached to the current node when the EXPAND mode is entered. The EXPAND mode is also in effect when CKTGEN is initially entered (default mode). As CKTGEN prompts you for component connections, the node expansion data for each pin is displayed. If you are not interested in this data, enter a NOEXPAND command to suppress the data.

A subsequent EXPAND command will reinstate the node expansion display. These commands require no arguments.

PINSALL: Display pin information from the DTL library for all pins on the current node. The characteristics (names, nail types, and pin types) for every pin on the node are displayed. The format is of the order: [RES C I] where RES are the first 3 letters of the node name; C is the nail type(s) currently allocated to this node, and I specifies the pin type e.g., I for input, O for output, TRI for 3-state, etc. (See also Figure 6-21 and the associated text Defining an IC.) The PINSALL command requires no arguments. Refer to NOPINSALL below for the alternate condition.

NOPINSALL: Display only the pin information for this IC (default condition). The information for the prompted pin is displayed in the form described in PINSALL above. If the extra information provided with PINSALL is not required, execution time is faster.

CONFIRM: Leave the prompt line unchanged after connection information is entered for a given pin. The prompt for the next pin then occurs only after 2 consecutive <RETURN>'s are entered (i.e., after a null entry). This allows you to examine the effect upon a node when a pin connection is entered. This command has no arguments.

NOCONFIRM: Move the prompt to the next pin after the first <RETURN> has been entered. This is the default condition (refer to the CONFIRM: command above).

MORE: The node connections are displayed in the NODE EXPANSION section of the component page display. If the entire node cannot fit in the one line of the display allocated for it, a plus (+) symbol appears to indicate an overflow exists. The MORE command is used to display the rest of the node/pins lines if the lines overflow. If there is no more overflow to display, the first part of the node is displayed again. The MORE command can take the form MORE, MORE NODE, or MORE NAILS.

6.8.3 GOTO Command

The GOTO command moves the prompt for a pin in the NODE CONNECTIONS section of a component page to another pin. The pin to be prompted for can be on any IC. There are several forms of GOTO as described below. The symbolic form ">" also described below is the most convenient one. You will likely use the > form of GOTO exclusively because it is the quickest. The other forms however, are equally valid on all CKTGEN pages.

arrows (up, down, left, right): Move cursor in direction of arrow. The prompt is changed to reflect the current cursor position. Left and right arrows only work in NODE CONNECTIONS area.

GOTO PIN=3: Move cursor to current component pin 3 and prompt for this pin connection. The node expansion display is for pin 3.

GOTO PIN=U14.3: Show component U14. The cursor is moved to pin 3, and the prompt and node expansion are for pin 3.

GOTO NAME=U14: Show component U14 and move the cursor to pin 1. The prompt and node expansion are for pin 1.

GOTO 3: CKTGEN interprets a number or single letter as a pin on the current component. The cursor is moved to the specified pin and the prompt reflects this move.

GOTO U14.3: CKTGEN interprets a name with a dot as a component pin. (This command is the same as GOTO PIN=U14.3)

GOTO U14: CKTGEN interprets a name as a component name. (This command is the same as GOTO NAME=U14)

>: same as GOTO (e.g., ">U14" is the same as "GOTO U14" which is the same as "GOTO NAME=U14"). (Note: this keystroke does not need to be preceded by the command mode # symbol.) This keystroke will prove to be the most useful of the GOTO forms.

> NOTE
>
> If any of the GOTO commands are performed within a DEFINE IC operation in a READ ONLY mode, then there is no prompt to enter a node connection.

6.8.4 SHOW Command

The SHOW command displays information in the NODE EXPANSION section of a component page. The expansion can be for a pin, a node, or a nail. There are several forms of SHOW as described below. The symbolic form "<" also described below is the most convenient one. You will likely use the < form of SHOW exclusively because it is the quickest. The other forms however, are equally valid on all CKTGEN pages.

SHOW PIN=3: Leave the cursor (and prompt) where it is now, but show the node expansion for pin 3.

SHOW PIN=U14.3: Leave the cursor (and prompt) where it is now, but show the node expansion for pin 3 of U14.

SHOW NODE=FOOBAR: Leave the cursor (and prompt) where it is now, but show the node expansion for the node named FOOBAR.

SHOW NAIL=F20: Leave the cursor (and prompt) at the current component and pin, but show the node expansion for the node with nail F20.

SHOW 3: CKTGEN interprets a number or single letter as a pin on the current component. This command is the same as SHOW PIN=3.

SHOW U14.3: CKTGEN interprets a name with a dot as component pin. This command is the same as SHOW PIN=U14.3.

SHOW F20: CKTGEN interprets a nail type followed by a number as a nail number. This command is equivalent to SHOW NAIL=F20.

SHOW FOOBAR: CKTGEN interprets a name that is not valid as a nail number as a node name. This command is equivalent to SHOW NODE=FOOBAR.

<: same as SHOW (e.g., "<U14.3" is the same as "SHOW PIN=U14.3"). (Note: this keystroke does not need to be preceded by the command mode # symbol.)

6.8.5 DEFINE, INSERT, Commands

CKTGEN references the system and update libraries to obtain pin count, pin name, nail requirement and other test information. If an IC does not exist in these libraries, it is defined and inserted in a user digital library (paragraph 6.7.4) From the definition, CKTGEN creates a test skeleton. Only digital IC components can be defined. The DEFINE command takes the form:

DEFINE TYPE=<dtl typename>: Display the DIGITAL IC DEFINITION page

(Figure 6-21). The type defaults to the current component type, if any, or else to EXT. This command is valid on all CKTGEN pages.

This command can be performed to examine an existing IC type. Additionally, if the IC being examined is located in a user digital library, it can be modified. ICs located in the other libraries are READ ONLY.

INSERT: Insert the currently-displayed type definition into the user DTL (create a test skeleton). This command requires no arguments.

6.8.6 REMOVE Command

This command is used to remove either a component, a pin connection, or a node that no longer has any pin connections on it. A REMOVE command requires a qualifier (an IC, IC pin, or node name) and takes the form:

```
REMOVE <component name>
REMOVE <component name>.<pin>
REMOVE <node name>
```

The REMOVE command causes an automatic VISIT (paragraph 6.8.7) to the node or nodes affected.

REMOVE U14: CKTGEN will prompt: "Removing whole IC, are you sure ? [Y,N]". You can only remove an IC which is currently displayed on the screen; the IC name is not optional. If you remove the last pin on any node, CKTGEN will prompt "Removing last pin on node. Remove IC, or ALL unconnected nodes and nails also? [NO, IC, ALL]"

REMOVE U14.6: Remove a pin connection. You can only remove a pin of an IC currently displayed on screen. If you are removing the last pin on a node, CKTGEN will prompt "Removing last pin on node. Remove node and its nails also? [Y,N]"

REMOVE FOOBAR: Remove a node with no pin connections and all of its nails.

6.8.7 Nail Allocation Commands

ADD Command

The ADD command allows you to join a 227x nail number to a node. You can thereby support more than one nail on a node, or accomodate circuit connections by adding a nail. The command first deletes the nail (refer to the DELETE command) from its nodal set before joining the nail to the new node. The form of the command is ADD n1=n2 where n1 is the new nail, and n2 is any existing test point on the node to receive n1. If the nail is to be added to the existing node, then the =n2 specification is not required. The effect of an ADD command can be seen in the node display in the NODE EXPANSION section of a component page.

ADD can be performed on a node regardless of its CAP/UNCAP status (described below) or its nail's tentative/unchangeable status. Nail numbers and types may be added as tentative or unchangeable. A tentative nail can be changed by CKTGEN if it finds a valid reason for doing so. CKTGEN will not disturb an unchangeable entity. A tentative nail type must be entered as a parenthesized entity. Entering a nail type with no number indicates a tentative nail number is desired. Entering a number with no type indicates a minimum capability nail type is desired. The nail type specifiers are: H (hybrid), F (fast), T (trigger), CD (clock driver), CS (clock sync), A (analog) and D (digital).

The following examples add to the node currently displayed in the NODE EXPANSION section of a component page.

ADD H12	type and number are unchangeable by CKTGEN
ADD (H)12	type is changeable, number is unchangeable by CKTGEN
ADD H(12)	type is unchangeable, number is changeable by CKTGEN
ADD (H12)	type and number are changeable by CKTGEN
ADD H	CKTGEN will assign the next available hybrid nail, the number is tentative.
ADD 12	CKTGEN will add a minimum capability nail, the type is tentative.

DELETE Command.

The DELETE command is used to delete a nail from the node currently displayed in the NODE EXPANSION section of a component page. DELETE can be performed on a node regardless of it's CAP/UNCAP status (refer to the CAP/UNCAP command), or its nail's tentative/unchangeable status (refer to the ADD command description for these aspects and the nail type specifiers). Parentheses are ignored with a DELETE command. Ambiguous commands (e.g., DELETE 12, where there is both a T12 and an H12 on the node) will result in an error message.

Examples of DELETE include:

DELETE H12	deletes hybrid nail 12 from the node currently displayed in the NODE EXPANSION section of the component page.
DELETE H	deletes a hybrid nail from the node currently displayed.
DELETE 12	Delete nail 12 from the currently displayed node.

The DELETE command is valid on a component page when a nail of the same type and/or number is displayed in the NODE EXPANSION section of the page.

DENOTE Command.

The DENOTE command changes the aspects (tentative/unchangeable) of both the type and number parts of a single nail on the current node. Tentative means that the CKTGEN nail allocation algorithm is allowed to change the type or number of a nail if it deems it desirable. The parethesization of the argument defines the resultant tentativeness in both type and number. A nail of the same type and number must be displayed on the current line. Both type and number are required for the command. DENOTE can be performed on nails regardless of their tentative/unchangeable or their node's CAP/UNCAP (described below) status. Examples of the tentative/unchangeable aspects and the nail type specifiers are detailed in the ADD command. CKTGEN will not alter the type or number of a nail if that type or number is marked unchangeable (no parentheses). DENOTE does not result in an automatic VISIT (described below).

Examples include: DENOTE H12 , DENOTE (H)12 , DENOTE H(12)

CAP/UNCAP Commands.

The CAP command marks a node(s) as CAPPED. This prevents CKTGEN from adding nails to, or deleting nails from the node(s). Tentative/unchangeable attributes of nails on the nodes are preserved. However, all nail numbers and types on capped nodes are displayed without parentheses, and the word "CAPPED" will appear on the nail line of the node display. This command is valid on any CKTGEN page.

There are 3 main forms of CAP. They are:

CAP NODE = N10 Prevent CKTGEN from adding nails to or deleting nails from the node(s). The keyword NODE = is optional. This form is the default condition.

CAP NAME = U21 Prevent CKTGEN from adding nails to or deleting nails from all nodes connected to component U21. The keyword NAME = is optional.

CAP BOARD Prevent CKTGEN from adding nails to or deleting nails from all nodes connected to this board. Note that the time to complete this operation depends on the number of nodes on the board.

UNCAP N10, UNCAP NODE = N10, UNCAP NAME = U21, UNCAP BOARD: Remove the CAP attribute from node N10, IC U21, or the entire board. UNCAP causes parentheses to reappear around tentative nail numbers and types. UNCAP does not cause an automatic VISIT (next command below). UNCAP is valid on any CKTGEN page.

VISIT Command

The VISIT command invokes the nail allocation algorithm. Nails are allocated or de-allocated as need to nodes. All nails assigned to the node are examined with regard to the nail requirements of pins connected to the node. CKTGEN will update the nail types of tentative nails and/or add new nails as long as the node is not CAPped. VISIT can take the following forms:

VISIT NODE N1 -- This specifies node N1. The keyword NODE is optional. This form is useful if the node has just been UNCAPped (described previously).

VISIT NAME U14 -- This specifies all nodes connected to pins on component U14. The keyword NAME is optional. This command is useful if the optional SPEEDPLUS capabilities are installed and this component requires FAST nails.

VISIT BOARD -- This specifies all nails. The keyword BOARD is required. This command is useful if a new target adaptor section has been added to a .CIR file and needs nails allocated, or if the optional SPEEDPLUS capabilities are installed and FAST nails need to be allocated.

The following actions cause an automatic VISIT to all affected nodes:

NOFAST flagspec
NOTEST flagspec
REMOVE
MOVE

Note that the commands ADD, DELETE, CAP/UNCAP, DENOTE, INSERT, and changing the target nail range do not cause an automatic VISIT.

A ^C (CTRL-C) command can be used to abort the visit process, however, this can result in incomplete nail allocations, requiring a subsequent visit command be performed.

6.8.8 Miscellaneous Commands

Precede each of the following commands with either a CTRL-G or with a # as described in paragraph 6.9.

TYPE = component type: Display the entry page for this component type and fill in the TYPE name. This command is valid on any page. The next prompt will be "ENTER NAME or #:" or, if the type is unknown to the libraries and is therefore a new type the prompt is: ENTER ANALOG OR DIGITAL (A/D) or #:".

NAME = component name: Display the component page for the named component if it exists, otherwise display an empty component entry page for the same type as currently displayed (the default type is EXTernal) and fill in the NAME. This command is valid on any page.

PINCOUNT=count: specify the number of pins on a component type not known to CKTGEN and display a template for pin connections. This entry will be automatically filled in by CKTGEN for ICs in the digital test libraries or standard analog types. This command is valid on component pages for types not known to CKTGEN or the DTL library. (Refer to para 6.7.4, Defining an IC.)

SAME = newname: where newname is a new component name. Enter a new component of the same type as the one currently displayed without changing any of the connections, values, or flagspecs. The name of the new component must be unique for the component being entered. If the name selected is already being used, the data corresponding to that name is retrieved and displayed. In any event, the current data is stored. For example, if a board contains 5 resistors R1-R3, R17 and R52 that are all of the same value, tolerance, you can enter the data for R1. Then, perform the command #SAME=R2. If any of the parameters, such as connections are different, you then GOTO (>) the pin you wish to change, and change it. WHen there are no more changes, perform the command #SAME=R3, and make any changes. Then perform the #SAME= commands for the remaining components R17, and R52.

MOVE n1=n2: Forms a union of all connections to 2 nodes and retains the name of the first node. For example, in the command MOVE n1=n2 n1 is any nail on one node, and n2 is any nail on the other node. The node name associated with n1 is retained. The MOVE command causes an automatic VISIT on the affected node. This command is valid on any page.

DEVICE = A or D: Specify device type (either analog or digital). This command is used if the component does not exist in a library accessed by CKTGEN. The command is valid on all component pages except for those with an elementary analog type. Changing the A/D type of a component which already has flagspecs assigned, causes CKTGEN to store the flagspecs as USER flagspecs.

NEWNAME = name: Change the name of the currently displayed component.

NEWTYPE = type: Change the type of the currently displayed component.

QUIT: Exit CKTGEN CREATE/REVISE without saving the updated .CIR file. This command is valid on any page. The message, "This session's data will be lost. Are you sure? [Y,N]:" will be displayed only if the user has changed the .CIR data base since entering CKTGEN. QUIT is useful for looking at data without creating a new .CIR file. A primary application for QUIT occurs if several errors have been introduced in the database file and you don't want to save it.

EXIT: Exit CKTGEN CREATE/REVISE, saves the current status including the CKTGEN monitor page, and board page information, and the .CIR database. This command is valid on any page.

GENERATE, GEN: Generate a .CKT file from the .CIR file. This command is valid on any page and converts the database into the file used in the nail assignment mode (NAS) of ATG. (Note that a ^C command can be used to abort this process before it completes. The .CIR file is saved.)

HELP: Display the CKTGEN help page. This command is valid on any page.

TARGET = Change the target tester used for displaying nail information, for the SHOW NAIL command, and for the Nail Allocation commands. This also affects the top line (header) of the page display of the target. If the .CIR file does not have the requested target nail data, CKTGEN will prompt:" To choose this target, return to the CKTGEN BOARD page by typing BOARD."

ERASE: This command applies only to the BOARD= and COMPONENT= flagspecs that are specified on the Board page (para 6.6.3) The effect is to delete a specific flagspec or all flagspecs listed. ERASE takes the forms:

#ERASE BOARD flagspec Erases a specific board flagspec. For example, if the Board page specified some board flagspecs as: BOARD=flagspec-1, flagspec-2, flagspec-3, and you want to erase flagspec-1 only, enter the command: #ERASE BOARD flagspec-1. (The brief form of the same command is #ERA B flagspec-1.)

#ERASE COMPONENT flagspec
Erases a specific flagspec from that component type only. For example if the Board page specified some component flagspecs as:
COMPONENT=R=RERR=1, MORERSPECS;
C=CERR=1, MORECSPECS; etc., and you want

to remove the R=RERR=1 flagspec, the
command to use is:
.#ERASE COMPONENT R=RERR=1.
The short form of this command is #ERA C
R=RERR=1.

CTRL C: This command (hold down the CTRL key while pressing the C) is used to abort the GENerate mode process of making a .CKT file from an existing .CIR database file. This ^C command is also used to abort a VISIT command including those command processes containing a visit within them ie. MOVE, NOTEST, NOFAST, REMOVE.

CTRL U: This command (hold down the CTRL key while pressing the U) erases the current line, leaving the prompt at the beginning.

6.8.9 CKTGEN Page Commands

REPORT: Display the CKTGEN REPORT file (.CRP) page. This command is valid on any page.

BOARD: Display the BOARD page and prompt for changes. This command is valid on any page.

6.9 COMMAND MODE

The commands described in the paragraphs of 6.8 and the CKTGEN HELP commands are executed by preceding the command with a sharp (#) symbol. CKTGEN requires the symbol to distinguish between a command and data entry by interpreting that which follows the # as a CKTGEN command. Alternately, a CTRL G command can be used instead of the #.

To leave the command mode enter one of the following commands or actions listed in Table 6-2.

Table 6-2
EXITING COMMAND MODE

Action or Command	Returns Prompt To:
up-arrow	Last prompt on page
down-arrow	First prompt on page
CTRL C	Previous prompt or monitor
CTRL H	Previous prompt on page
CTRL N	Next prompt on page
CTRL T	First prompt on page
CTRL Z	Last prompt on page
GOTO (para 6.8.3)	Pin on - this page - different page - new page
off-page (e.g. TYPE=,	New page NAME=, etc.)
<RETURN>	(return key not preceded by a command)

Some keystroke commands including most of those listed in Table 6-5 can be executed without entering the command mode ie., don't need to be preceded by a CTRL G or a # symbol. These commands include:

CTRL C	Abort visit process or abort GENerate mode only.
CTRL H	Set the prompt to the previous prompt.
CTRL L	Rewrite the screen, do not change the prompt.
CTRL N	Set the prompt to the next prompt.
CTRL T	Set the prompt to the first option on the page.
CTRL U	Erase the current line, leaving the prompt at the beginning.
CTRL Z	Set the prompt to the last option on the page.
>	GOTO (para 6.8.3).
<	SHOW (para 6.8.4).
(null entry followed by a <RETURN>)	Set the prompt to the next prompt.
arrows (up, down, right, left)	Move the cursor in the direction of the arrow and prompt. Left and right arrows work only in the NODE CONNECTIONS area. Arrows are disabled on the DEFINE page in READ ONLY mode, as displayed data cannot be changed.
<PF4> key or <HELP> key	Display a 2-line description of the current prompt.

6.10 CIRCUIT FILE READER

An existing circuit description can be modified by running the CKTGEN READ mode. On the monitor page, the user can specify:

an input .CKT file	
an output .CIR file	
a .ERC error-listing file	
a DTLusr library file (DTLusr=)	for new or existing ic's not currently contained in the system digital libraries.
a device for scratch files (Temp=)	used when new ic's not known to any digital library are defined, or when an ic definition in the user library is to be modified.

This takes the existing .CKT file and creates a .CIR database file from it. The newly created .CIR file can then be revised. This operation should not be confused with revising an existing .CIR file. If an existing .CIR file is known to have been created using the current software release, it can be REVISEd, not

READ; otherwise, the .CKT file should be READ first. A read operation allows a user to modify a .CKT file, eliminating the need for the user to archive the .CIR file to continue to use CKTGEN. This is especially important for multiplexed testers where nail assignment modifies the .CKT file and renders the user's .CIR file obsolete.

The new .CIR file will store BOARD page (target) information along with the circuit description. The circuit file reader determines which targets are active and the nail counts from each active target from the %ADAPTOR section of the .CKT file.

Considerations

1. .CKT files produced by CKTGEN in earlier software releases often contain %CIRCUIT entries of the form:

```
N1   EXT   R1.2,U10.1;
```

where N1 is a node identifier rather than an external connection. If such statements occur, the circuit file reader treats N1 as though it was an external connection, and will create an additional node identifier. Therefore, it is recommended that such entries be moved from the %CIRCUIT section to the comment preceding it, or deleted from the .CKT file by manually editing before reading the .CKT file.

2. The input .CKT file is handled by the circuit file reader in much the same way as circuit preparation. Note however, that circuit preparation ignores %ADAPTOR data applicable to target testers other than the target selected in the SETUP mode. Hence, is TARGET=2271 is chosen for ATG, and syntactic or cross-connectivity errors occur in another adaptor section, the errors will not be detected in circuit preparation. Since the circuit file reader processes all adaptor sections, the errors will prevent the .CKT file from being read successfully.

3. Any .CKT file warnings or errors are listed in the .ERC file; however, CKTGEN will still attempt to create a .CIR file.

4. The circuit file reader will accept incomplete .CKT files; however, an existing %CIRCUIT section is required to produce a .CIR file. A warning message is issued if any .CKT-file section is not found.

5. For an existing .CKT file, there is a comment block at the top of the this file which is read in on a READ command and output for a GENerate command. This comment block can not be modified by a CREATE/REVISE operation. Other comments are not retained during a READ command.

6. Every circuit node has a node name, either provided by the user or created by the .CKT File Reader. (A node name is the name of a connection on an IC pin. Device pins for which the user has entered no connection information have no associated node.

7. A .ERC file can be specified on the CKTGEN monitor page. This file contains error information similar to that found in the .ERR listing produced during circuit preparation in ATG.

6.11 TRAINING BOARD PARTS CIRCUIT DESCRIPTION

6.11.1 General

Figure 6-22 is a logic diagram of the 2275-4715 Training Board. This board is used as an example throughout the system documentation and as a training tool at the GenRad Training facility to teach you the process of test set preparation. A .CKT circuit description file for this board has been compiled for use as a reference using CKTGEN. If you are going to duplicate this task, the .CKT file can be used as a parts list when the database (.CIR file) is CREATEd.

The node connections (temporary nails) shown on the schematic are in reality the final nail assignments. Because GenRad has already manufactured the fixture, that process required that permanent nails be assigned (Nail Assignment Mode). These final nail assignments are used here as though they are initial (temporary) assignments. In practice, you will first annotate the diagram, perform CKTGEN and the rest of ATG and obtain final nail assignments to be used in the manufacture of the fixture.

6.11.2 Creating the Circuit Description

The circuit description for the Training Board was created using the following CKTGEN page specifications and circuit considerations:

1. The Setup page specifications are: NAME=TRAININGBD , DEV=SC: , and the 2275 is the target system.

2. The CKTGEN Monitor page specifications are: MODE=CREATE , DTLusr=Y (and an appropriate user library specified).

3. The CKTGEN Board page specifications are: TARGET 2275 [N,Y]=Y . Additionally, for this target, the Hybrid Nail Count=256, Probe=64 and Sync=63 are entered.

4. The capacitors C8-C20 are not inserted because they are power supply capacitors that are not tested.

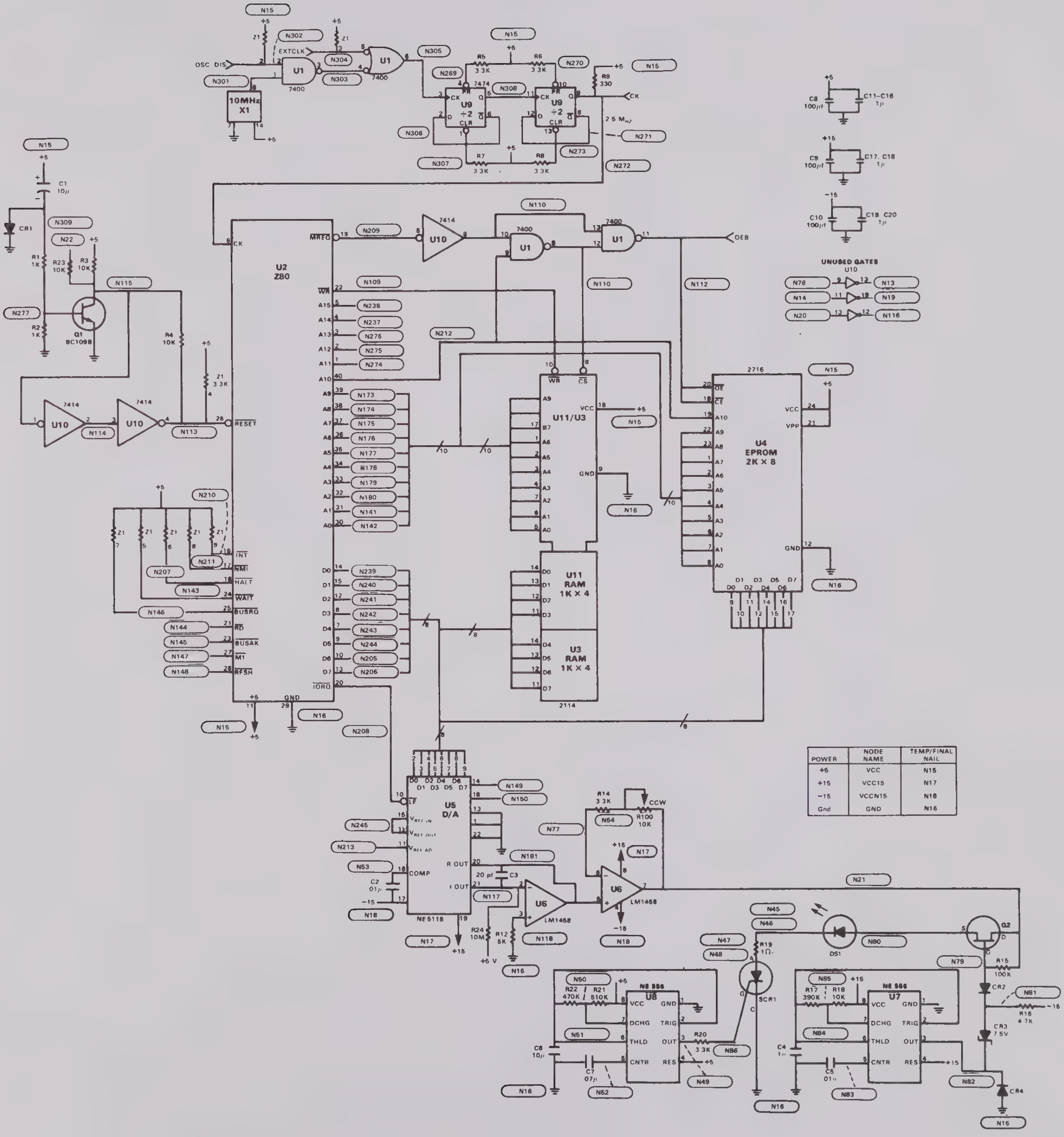
5. R19 is a Kelvin (4-terminal) connection. Refer to the ADD command to add the second set of nails.

6. Component Z1 is a resistor pack (8 resistors and 9 pins). This is a single 9-pin component. Pin 1 is connected to VCC. It is created by entering the command #TYPE=RPAK8 followed by a #DEV=ANALOG command.

7. Crystal Y1 is inserted as a 14-pin device even though it has only 3 pins. This will accomodate the unique numbering for these 3 pins (Pins 7,8,14).

8. The 2114 RAMs U3 and U11 have many pins in common. After U3 is entered U11 can be entered using a #SAME=U11 command and then changing only pins 11-14 for U11.

9. The digital type NE5118 (D/A Converter) is not known by the digital libraries. Refer to the #DEFINE TYPE and #INSERT commands. (Note that the connections to U5 can be made before the ic is defined using a #TYPE=NE5118 command.)


OSC DIS
EXTCLK
10MHz X1
U1 7400
U9 ÷2 7474
R5 3.3K
R6 3.3K
R7 3.3K
R8 3.3K
R9 330
CK
2.5 MHz
C1 10μ
CR1
R1 1K
R2 1K
R3 10K
R23 10K
R4 10K
Q1 BC109B
U10 7414
U2 Z80
MREQ
WR
RESET
INT
NMI
HALT
WAIT
BUSRQ
RD
BUSAK
M1
RFSH
IORQ
U11/U3
U11 RAM 1K × 4
U3 RAM 1K × 4
2114
U4 EPROM 2K × 8
2716
OE
CE
VCC
VPP
GND
OEB
U5 D/A
NE5118
LE
COMP
R OUT
I OUT
C2 .01μ
C3 20 pf
U6 LM1458
R14 3.3K
R100 10K
CCW
R24 10M
R12 5K
R19 1Ω
SCR1
DS1
Q2
R15 100K
CR2
CR3 7.5V
CR4
R16 4.7K
U8 NE 555
U7 NE 555
VCC
GND
DCHG
TRIG
THLD
OUT
CNTR
RES
R22 470K
R21 510K
R17 390K
R18 10K
R20 3.3K
C6 10μ
C7 .07μ
C4 1μ
C5 .01μ
C8 100μf
C11–C16 1μ
C9 100μf
C17, C18 1μ
C10 100μf
C19, C20 1μ
UNUSED GATES U10

POWER	NODE NAME	TEMP/FINAL NAIL
+5	VCC	N15
+15	VCC15	N17
−15	VCCN15	N18
Gnd	GND	N16

Figure 6-22. Training Board Logic Diagram.

6.11.3 Training Board .CKT File

Once the database file (.CIR) has been created, the circuit description can be made from the CKTGEN Monitor page using the MODE=GEN (for generate) option and running the mode.

The following listing of the .CKT file is supplied as a reference.

```
/* SC:TRAINGBD.CKT Created by GenRad Circuit Generator */

%CIRCUIT;

GND       EXT       ,      N16;
VCC       EXT       ,      N15;
VCC15     EXT       ,      N17;
VCC15N    EXT       ,      N18;

C2        C         N53, N18;
C3        C         N181, N117;
C4        C         N84, N16;
C5        C         N83, N16;
C6        C         N51, N16;
C7        C         N52, N16;

C1        CP1       P = N15, N = N309;

CR1       CR        A = N309, C = N16;
CR2       CR        A = N79, C = N81;
CR4       CR        A = N16, C = N82;
DS1       CR        A = N80, C = N45;

Q1        QN        B = N277, C = N115, E = N16;

R1        R         N309, N277;
R12       R         N118, N16;
R13       R         N77, N16;
R14       R         N77, N54;
R15       R         N21, N79;
R16       R         N18, N81;
R17       R         N85, N84;
R18       R         N17, N85;
R19       R         N45, N47;
R2        R         N277, N16;
R20       R         N86, N49;
R21       R         N15, N50;
R22       R         N50, N51;
R23       R         N22, N115;
R24       R         N15, N117;
R3        R         N15, N115;
R4        R         N115, N113;
R5        R         N15, N269;
R6        R         N15, N270;
R7        R         N15, N307;
R8        R         N15, N273;
R9        R         N15, N272;

R100      RV        N54, N54, N21;

CR3       VZ        A = N81, C = N82;

Q3        SCR       A = N47, C = N16, G = N86;

Q2        NJFET     G = N79, D = N80, S = N21;

U6        LM1458    N181, N117, N118, N18, N181, N77, N21, N17;
Z1        RPAK8     N15, N302, N304, N113, N143, N207, N146, N211,
                   N210;


U1        7400      N301, N302, N303, N303, N304, N305, N16, N110,
                   N212, N111, N112, N110, N111, N15;

U10       7414      N115, N114, N114, N113, N209, N111, N16, N13, N78,
                   N19, N14, N116, N20, N15;

U11       2114      N176, N177, N178, N179, N142, N141, N180, N110,
                   N16, N109, N206, N205, N244, N243, N173, N174,
                   N175, N15;

U2        Z80CPU    N274, N275, N276, N237, N238, N272, N243, N242,
                   N244, N205, N15, N241, N206, N239, N240, N210,
                   N211, N207, N209, N208, N144, N109, N145, N143,
                   N146, N113, N147, N148, N16, N142, N141, N180,
                   N179, N178, N177, N176, N175, N174, N173, N212;

U3        2114      N176, N177, N178, N179, N142, N141, N180, N110,
                   N16, N109, N242, N241, N240, N239, N173, N174,
                   N175, N15;
```

```
U4        2716      N175, N176, N177, N178, N179, N180, N141, N142,
                   N239, N240, N241, N16, N242, N243, N244, N205,
                   N206, N112, N212, N112, N15, N173, N174, N15;

U5        NE5118    N16, N239, N240, N241, N242, N243, N244, N205,
                   N206, N208, N213, N245, N16, N149, N245, N53, N18,
                   N150, N17, N181, N117, N16;

U7        NE555     N16, N84, N82, N17, N83, N84, N85, N18;

U8        NE555     N16, N51, N49, N15, N52, N51, N50, N15;

U9        7474      N307, N306, N305, N269, N308, N306, N16, N271,
                   N272, N270, N308, N271, N273, N15;

Y1        XTL       7 = N16, N301, 14 = N15;

%VALUE;


C2         = 10.0N, +80%,-20%;
C3         = 20.0P, 5%;
C4         = 1.0U, 10%;
C5         = 10.0N, +80%,-20%;
C6         = 10.0U;
C7         = 10.0N, +80%,-20%;

C1         = 10.0U, 20%;


R1         = 1.0K, 5%;
R12        = 5.1K;
R13        = 10.0K, 5%;
R14        = 3.3K, 5%;
R15        = 100.0K, 5%;
R16        = 4.7K, 5%;
R17        = 390.0K, 5%;
R18        = 1.0MEG, 5%;
R19        = 1, 1%;
R2         = 1.0K, 5%;
R20        = 3.3K, 5%;
R21        = 510.0K, 5%;
R22        = 470.0K, 5%;
R23        = 10.0K;
R24        = 10.0MEG, +5%, -5%;
R3         = 10.0K, 5%;
R4         = 10.0K, 5%;
R5         = 3.3K, 5%;
```

```
/* SC:TRAINGBD.CKT Created by GenRad Circuit Generator */



H174   = N174;       H175   = N175;       H176   = N176;
H177   = N177;       H178   = N178;       H179   = N179;
H180   = N180;       H181   = N181;       H205   = N205;
H206   = N206;       H207   = N207;       H208   = N208;
H209   = N209;       H210   = N210;       H211   = N211;
H212   = N212;       H213   = N213;       H237   = N237;
H238   = N238;       H239   = N239;       H240   = N240;
H241   = N241;       H242   = N242;       H243   = N243;
H244   = N244;       H245   = N245;       H269   = N269;
H270   = N270;       H271   = N271;       H272   = N272;
H273   = N273;       H274   = N274;       H275   = N275;
H276   = N276;       H277   = N277;       H301   = N301;
H302   = N302;       H303   = N303;       H304   = N304;
H305   = N305;       H306   = N306;       H307   = N307;
H308   = N308;       H309   = N309;

#PROBE = H64;        #SYNC  = H63;
```

```
/* SC:TRAINGBD.CKT Created by GenRad Circuit Generator */


R6         = 3.3K, 5%;
R7         = 3.3K, 5%;
R8         = 3.3K, 5%;
R9         = 330.0, 5%;

R100       = 10.0K, 20%;

CR3        = 7.5, 5%, MSG='7.5 ZENER DIODE';


U6         = POSRAIL=15, NEGRAIL=15;

Y1         = NOTEST;


%ADAPTOR : 2275 TEMP;



H13    = N13;        H14    = N14;        H15    = N15;
H16    = N16;        H17    = N17;        H18    = N18;
H19    = N19;        H20    = N20;        H21    = N21;
H22    = N22;        H45    = N45;        H46    = N45;
H47    = N47;        H48    = N47;        H49    = N49;
H50    = N50;        H51    = N51;        H52    = N52;
H53    = N53;        H54    = N54;        H77    = N77;
H78    = N78;        H79    = N79;        H80    = N80;
H81    = N81;        H82    = N82;        H83    = N83;
H84    = N84;        H85    = N85;        H86    = N86;
H109   = N109;       H110   = N110;       H111   = N111;
H112   = N112;       H113   = N113;       H114   = N114;
H115   = N115;       H116   = N116;       H117   = N117;
H118   = N118;       H141   = N141;       H142   = N142;
H143   = N143;       H144   = N144;       H145   = N145;
H146   = N146;       H147   = N147;       H148   = N148;
H149   = N149;       H150   = N150;       H173   = N173;
```

6.12 CIRCUIT DESCRIPTION LANGUAGE (CDL)

The following paragraphs describe the Circuit Description Language only; for information on how to create an analog component model for insertion into a user library, please refer to the Test Library Programming Manual, Section 7.

6.12.1 General Guidelines

The following general guidelines apply to the coding of the circuit description. More detailed information is contained in subsequent paragraphs.

- Each CDL statement must be terminated by a semi-colon (;).
- If the number of entries in a statement exceeds the 80-character per line capacity of the display, the statement may be extended to more than one line. The statement end is denoted by a semi-colon (;) and not by a return.
- Comments may be included in any statement in either of 2 formats:

  All characters following a slash (/) to the end of the line are interpreted as a comment.

  All characters bracketed by /* Character String */ are interpreted as a comment, regardless of end-of-line.

NOTE

Comments may be nested (i.e., comments may be specified within a comment string). This is used when some statements are included for reference but are not to be executed.

6.12.2 Circuit Description File Format

The circuit description is coded and structured in a table-oriented format which uses the following 3 types of data specifications:

CIRCUIT - Specifies the name, type, and circuit interconnections for each component or external node.

VALUE - Specifies value and tolerance for circuit components. May also include optional flagspecs defining other component parameters or special operator messages.

ADAPTOR - Specifies the circuit connections to the system test nails.

The CIRCUIT section must precede the VALUE and ADAPTOR sections in the circuit description file. Headers are used to identify each section. The 3 sections of data should be organized as shown in Figure 6-23. The following paragraphs describe each of these 3 sections, and include syntax, format and usage guidelines. For reference, a sample .CKT file (developed by CKTGEN) is listed in paragraph 6.11.

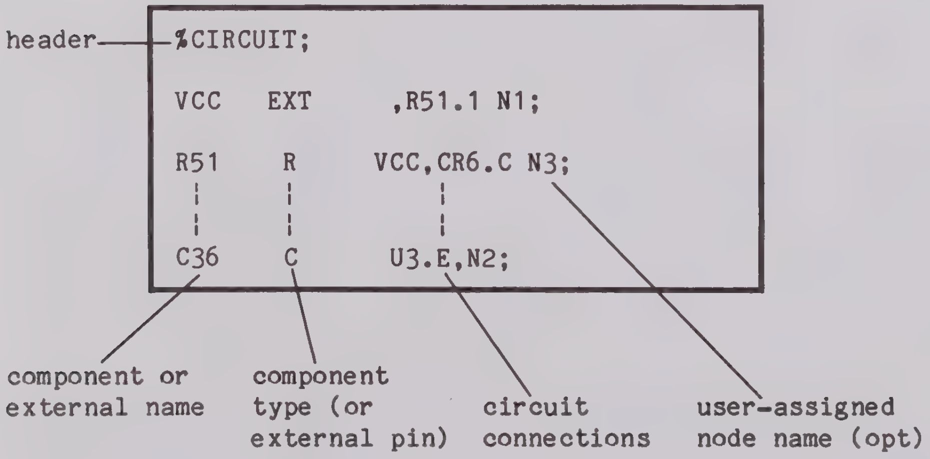
```
/* Comment that identifies the pc board */

%CIRCUIT;

      component-name  type   circuit-connections;


%VALUE:227x;

      component-name(s) = value,tolerance,flagspecs;


%ADAPTOR:2270;  or  %ADAPTOR:227x TEMP;

      test-nail(s) = circuit-node(s);
```

Figure 6-23. Format of .CKT File.

6.12.3 %CIRCUIT Section

The following example illustrates the format and content of a typical %CIRCUIT section. Additional examples appear at the end of this paragraph.

NOTE
Each component and external connection point on the pc board must have a CIRCUIT statement.

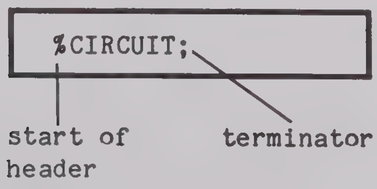


```
%CIRCUIT;

VCC   EXT     ,R51.1 N1;

R51    R     VCC,CR6.C N3;
 |     |          |
 |     |          |
C36    C       U3.E,N2;
```

header
component or external name
component type (or external pin)
circuit connections
user-assigned node name (opt)

The section must start with the following header:

Header

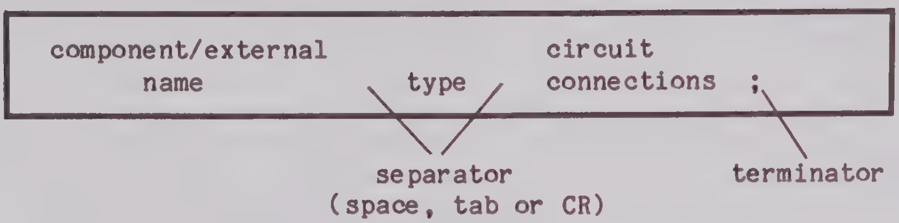


```
%CIRCUIT;
```

start of header
terminator

The CIRCUIT statements that follow this header all have 3 distinct fields of information.

Circuit statement



The name field specifies either a circuit component or an external node (i.e., a circuit point that is an external connection to the board). The type field specifies the type of component or node. The connections field defines connections to each component lead (or external node) specified in the name field. This field can also be used to assign a node name (or intermediate name) to each circuit connection.

A more detailed description of each of these fields follows:

A. Name Field

- Component and external names are user-assigned, and typically taken from the schematic diagram (e.g., R51, C36, VCC).

- These user-assigned names are made up of 8 (max) alphanumeric characters which include letters A-Z, the numbers 0-9 and the underline (_). No space or other character is allowed. Each name must be unique. CDL will truncate names that have more than 8 characters and will issue a warning. The warning alerts the user to the fact that the name is different from that specified or that a delimiter may have been omitted from the end of the name.

- Power (VCC) and ground (GND) nodes must be properly defined for digital/hybrid tests. These power nodes (VCC and GND) can either be declared as EXTernals in the type field or can be assigned as node names (intermediate names) in the connection field. For boards with multiple power nodes, additional characters can be added to VCC and GND for further identification (e.g., VCCP5, GNDM15, etc). ATG only looks at the first 3 characters and interprets VCCxxx as logic 1, and GNDxxx as logic 0.

- Digital ICs are considered as single components by the digital test library. Only one name may be used to define a digital IC, even though there may be several independent circuits within the package. If the manufacturer's type has more than 8 characters, use the system-assigned name, which can be found in the library directory.

B. Type Field

- Component types must be specified just as they appear in Table 6-3 (e.g., R, C, QN, etc).

- All external connections must be specified as type EXT. To provide compatibility with other GenRad systems, however, the compiler will also accept keywords EXTIN and EXTOUT as meaning EXT. Externals that enter the board and do not make contact with any board components should be coded in as a JUMPER. This assures that the .NFR and .NAR files reflect where the receiver nails are wired to the fixture nails.

- Semiconductor devices that are not defined in Table 6-3 and which cannot be approximated by device types QP, QN, VZ, or CR should still be included in the circuit description as IC types. This allows ATG to insert a generalized back-to-back diode model in place of the undefined device, when evaluating the equivalent circuit.

- ICs which do not have a test in the libraries must be declared as type ICnn, DICnn, or UKnn. ICnn provides information for analog ATG circuit analysis. ATG will default to IC40 for any component not defined by the user. If pin data is required in the .IDD file for an untested device, the DICnn must be specified.

C. Connection Field

NOTE

Test fixture nails are not specified in this field.
Those connections are defined in the %ADAPTOR section.

- Pin connections to each component lead (or node) are specified by the system-defined pin identifiers shown in Table 6-3 and cannot be arbitrarily assigned by the user. These pin identifiers are either letters or numbers that define a component lead, for example,

R51.1 indicates a connection to pin 1 of resistor R51, or
CR6.A indicates a connection to the anode of diode CR6.

- For components with numbered pins, the first connection starts with pin 1 and increases sequentially through all the pins of that component. Commas separate the pin connections and each comma increments the pin count by one. At any pin count (including pin 1), this automatic pin sequencing can be interrupted and an = sign used to define a specific pin. After this break, commas would then continue to increment the pin count by one, starting at the new number. For example, the statement

```
U3    IC     CR3.A,3=C31.2 R14.1,R23.1,,R56.2;
```

would indicate the following circuit connections to U3:

Pin 1 of U3 connects to pin A of CR3
Pin 2 of U3 is not connected
Pin 3 of U3 connects to pin 2 of C31 and pin 1 of R14
Pin 4 of U3 connects to pin 1 of R23
Pin 5 of U3 is not connected
Pin 6 of U3 connects to pin 2 of R56

Table 6-3
COMPONENT TYPE SYMBOLS USED IN CDL

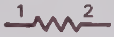
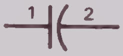
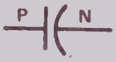
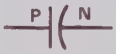
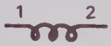
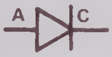
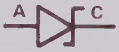
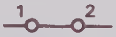
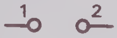
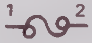
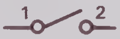
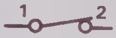
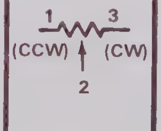
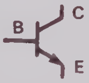
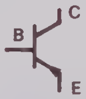
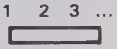
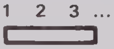
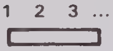
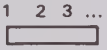
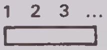
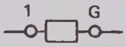
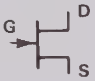
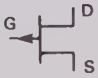
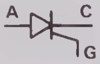
COMPONENT TYPE	CDL SYMBOL	SCHEMATIC SYMBOL	COMPONENT DESCRIPTION
RESISTOR	R		Any resistor. The leads must be designated as 1 and 2.
CAPACITOR	C		Any non-polarized capacitor. The leads must be designated as 1 and 2.
CAPACITOR POLARIZED	CP		Polarized capacitor. The leads must be designated P for the positive and N for the negative. No reverse bias greater than 1.0 volts is allowed by ATG.
CAPACITOR POLARIZED	CP1		Same as CP except no reverse bias greater than 0.0 volts is allowed by ATG. Typically for tantalum capacitors.
INDUCTOR	L		Any inductor. The leads must be designated as 1 and 2.
DIODE	CR		Any non-zener diode. The leads must be designated A for anode and C for cathode.
ZENER DIODE	VZ		Any zener diode. The leads must be designated A for anode and C for cathode.
JUMPER	J		Any very low resistance connection. The ends must be designated 1 and 2.
NO JUMPER	OJ		Any open circuit. The nodes must be designated 1 and 2.
FUSE	F		Any fuse. The leads must be designated 1 and 2.
OPEN CONTACT	NO		Any normally open contact. The terminals must be designated 1 and 2.
CLOSED CONTACT	NC		Any normally closed contact. The terminals must be designated 1 and 2.
POTENTIOMETER	RV		Any adjustable resistor. The leads must be designated such that the wiper is 2, the terminal that the wiper approaches when turned counter clockwise is 1, and the terminal that the wiper approaches when turned clockwise is 3.
TRANSISTOR NPN	QN		Lead designations are B for base, C for collector, and E for emitter.
TRANSISTOR PNP	QP		Lead designations are B for base, C for collector, and E for emitter.

Table 6-3 (continued)
COMPONENT TYPE SYMBOLS USED IN CDL

COMPONENT TYPE	CDL SYMBOL	SCHEMATIC SYMBOL	COMPONENT DESCRIPTION
PC BOARD	BOARD		Used when a flagspec that applies to the pc board is to be specified by the user.
DEVICES NOT RECOGNIZED BY ATG LIBRARIES	ICnn		A device for which the ATG libraries do not have a test. Assumed to be a semiconductor device; so, the diode model is used, the VCC and GND guarding is used and the knee voltage VETO is used. The number of pins on the device replaces the nn in the CDL symbol. The number of pins allowable is any integer from 2 through 127. Default is 40 pins.
	ICAnn		Same as ICnn except that the guarding is not done.
	DICnn		A digital IC which has no test in the library but ATG will provide a dummy burst in the test program and nail assignment will reserve pins on mux tester. The number of pins allowable is 8 or an even number from 14 through 40.
	UKnn		A device that is assumed not to be a semiconductor device, so no guarding of VCC and GND, no knee voltage VETO, and no diode model are used. The impedance between pins is considered to be infinite for ATG circuit analysis. Pin quantity is same as ICnn.
INTEGRATED CIRCUIT	(type)		An IC for which there is a test in a digital library or component library. Pin numbers are consistent with manufacturers data and consequently with library specification.
USER SPECIFIED GUARD	GD		Forces a guard at node G regardless of whether the library procedure specifies a guard.
COMPONENT LIBRARY CIRCUIT ELEMENTS			
FIELD EFFECT TRANSISTOR N JUNCTION	NJFET		Lead designations are D for drain, G for gate and S for source. ATG provides a FORM test only for this device. The user must complete the test.
FIELD EFFECT TRANSISTOR P JUNCTION	PJFET		Lead designations are D for drain, G for gate and S for source. ATG provides a FORM test only for this device. The user must complete the test.
SILICON CONTROLLED RECTIFIER	SCR		Lead designations are A for anode, C for cathode, and G for gate.

- Note that multiple connections to a node are separated by a space. For example

```
R16  R  R14.1,U3.C C35.1 R24.1;
```

indicates that pin 2 of R16 is connected to 3 other component leads: U3.C, C35.1 and R24.1.

- For components with lettered pins, typically polarized components such as transistors and diodes, pin sequencing is never automatic and each pin must be explicitly assigned using the = sign. For example, the base, emitter and collector connections to a transistor would be specified as follows:

```
Q3    QN     B=R23.1 C34.2,E=GND,C=R22.2;
```

- For EXTernal type connections only, a comma must precede the first entry. For example,

```
VCC   EXT     ,R18.1 R7.2;
```

This restriction exists only for compatibility with other GenRad circuit board test systems. The program ignores any connection before the first comma.

- The first time a node or component connection appears in a circuit statement, list all connections to that node (e.g., IC6.1, R3.1, C6.2). Thereafter, when that node appears again in a statement, the compiler needs only one of those connections to be specified (e.g. R3.1). Additional references will cause the compiler to cross-check the original connection (at the cost of additional compiling time).

Therefore, to take advantage of this cross checking feature while still avoiding excessive repetitions in the source file and increase in compile time, it is recommended that only one additional cross-reference be made to each connection.

- To minimize the risk of making wrong connection entries, node connections should be checked off on the schematic as they are entered. To facilitate ATG's cross-checking of the entries, different connections should be used on each subsequent reference to the node.

- At least one connection must be specified for each lead of an analog component defined in a CIRCUIT statement (i.e., an analog component that appears in the name field of that statement).

- The user can assign a node name to each circuit node by specifying that name in the connection field. For example,

```
R51  R  VCC,CR6.C N3;
```

assigns node name N3 to the connection point where R51 pin 2 is attached.

- User assigned node names should not be of a form that could be interpreted as a nail identifier, eg. "H101". The common practice of "N" followed by an integer is conformed to by the circuit file reader.

- Node names can be used as "intermediate" names to merge two or more points on a schematic into a single node. In instances where a circuit node extends over several pages of a schematic diagram, assigning an intermediate name to that node allows the individual pages to be coded separately.

D. CIRCUIT Statement Examples

1. `R53  R  Q3.C L3.2 CR3.A,C7.2;`

Meaning: Specifies that component R53 is a component type R (resistor) and that pin 1 of this component is connected to components Q3 pin C, L3 pin 2, and CR3 pin A. Pin 2 of this component is connected to component C7 pin 2.

2. `VCC  EXT ,CR4.C R22.1 C33.2;`

Meaning: Specifies that external connection VCC has components CR4 pin C, R22 pin 1, and C33 pin 2 connected to it.

3. `U9 7451 D1,,,,,, GND, OUT, D4, D3,,, D2, VCC;`

Meaning: U9 is a dual gate, with each half of the IC tested separately. In this application, only one gate is used in the circuit. This example specifies U9 as a type 7451 IC with pins 1, 7, 8, 9, 10, 13, 14 being those connected for one of the gates. The pins for the unconnected gate are those that exist between the multiple commas in the sequence.

4. `U6 DIC8 IN1, IN2, IN3, IN4, GND,, OUT, VCC;`

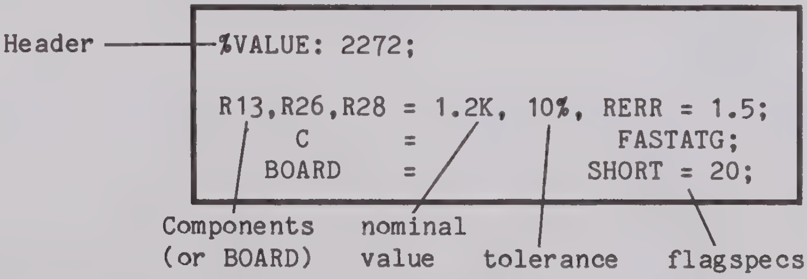
Meaning: U6 is an 8-pin device type that does not have a library procedure. DIC8 is specified so that the dummy 8-pin library procedure can be used to provide connection data for the .IDD file that is used at run time.

5. `U3 IC 7=GND, 14=VCC, 20=R3.1, 22=R4.2;`

Meaning: This is an IC that has only four of its pin connections defined and which does not have a digital library procedure. ATG will default the IC to a 40-pin device and model the defined pins for circuit analysis purposes. No .IDD data is provided.

6.12.4 %VALUE Section

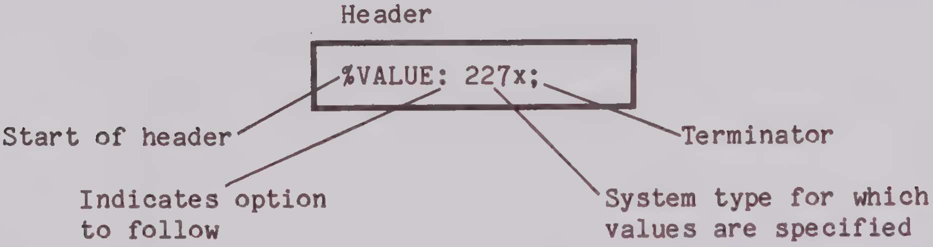
The following example illustrates the format and content of a typical %VALUE section. Additional examples appear at the end of this paragraph.



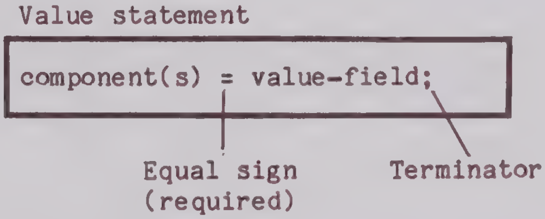
```
%VALUE: 2272;

R13,R26,R28 = 1.2K, 10%, RERR = 1.5;
    C       =             FASTATG;
    BOARD   =            SHORT = 20;
```

NOTE
The spacing in this example has been included for clarity only; it is not required in the format.

The %VALUE section must start with the following header statement:


Header
%VALUE: 227x;
Start of header
Terminator
Indicates option to follow
System type for which values are specified

The VALUE statements that follow this header consist of 2 distinct fields, one on each side of the equal (=) sign.


Value statement
component(s) = value-field;
Equal sign (required)
Terminator

The *component-field* identifies all components or component types characterized by the entries in the value field. Entries in the component field can specify individual components (e.g., R13, R26, R28), component types (e.g., R, C, RPAC, etc), or all components on the board (by using the keyword BOARD).

The *value-field* defines the nominal value and tolerance of components in the component field, and can also assign special flagspecs to these components. Table 6-3 summarizes some of the parameter information for the value field.

A more detailed description of both of these fields follows.

A. *Component Field*

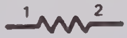
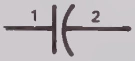
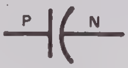
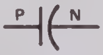
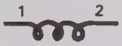
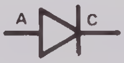
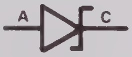
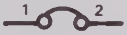
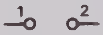
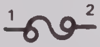
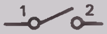
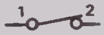
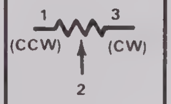
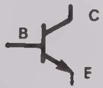
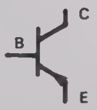
- Entries in this field specify the individual or collective components that are characterized by the entries in the value field. Note that the system assigns to *all* components in this field the characteristics defined by *all* parameters in the value field.

- Components can be specified in any 1 of 3 ways:

  1. Individual components such as R13,R26,R28, etc. Each of these components must have previously appeared in the name field of a %CIRCUIT statement. If multiple components are entered in this field, they must be separated by commas.

  2. Component types such as R, C, QN, etc. Component type entries are allowed providing that all parameters in the value field apply to all components of the type specified.

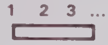
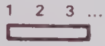
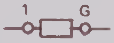
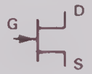
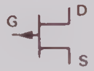
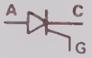
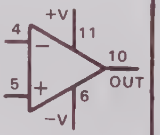
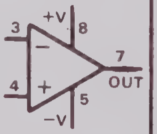
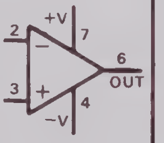
Table 6-4
CIRCUIT DESCRIPTION PROGRAM ELEMENTS

DEVICE TYPE	CDL SYMBOL	PIN ASSIGNMENT	VALUE SECTION SPECIFICATIONS			
			NOMINAL CHARACTERISTIC VALUE	TOLERANCE	IMPLEMENTED TEST LIBRARY OPTIONAL FLAGSPECS	
BASIC CIRCUIT ELEMENTS						
RESISTOR	R		RESISTANCE (OHMS)	[5%]	RHIRESID [3.2 ohms] RLORESID [1.8 ohms] ILIM [.128]	RERR [1.0] RLOV [500 ohms] RHIV [1 MEG] FASTATG
CAPACITOR (NON-POLARIZED)	C		CAPACITANCE (FARADS)	[5%]	CHIRESID [20 pF] CLORESID [5 pF] FREQ [System-optimized]	CERR [1.0] FASTATG
CAPACITOR (POLARIZED, UP TO 1 V REVERSE VOLTAGE)	CP		CAPACITANCE (FARADS)	[20%]	CHIRESID [20 pF] CLORESID [5 pF] FREQ	CERR [1.0] FASTATG
CAPACITOR (POLARIZED, NO REVERSE VOLTAGE PERMITTED)	CP1		CAPACITANCE (FARADS)	[20%]	CHIRESID [20 pF] CLORESID [5 pF] FREQ	CERR [1.0] FASTATG
INDUCTOR	L		INDUCTANCE (HENRYS)	[20%]	LERR [1.0] DCR [1 ohm] FREQ [System-optimized] RTOL [No Resistance Test]	LHIRESID [15 μH] LLORESID [5 μH] FASTATG
DIODE	CR		KNEE VOLTAGE (VOLTS) [.65]	[30%]	DR [*] DRH DRL IFV [20 mA] IDR [5 mA]	FVONLY
ZENER DIODE	VZ		ZENER VOLTAGE (VOLTS)	[10%]	IZT [(7.8 x 16/VZ) mA] ZZT [(17 x VZ/16) ohms]	
JUMPER	J		RESISTANCE (OHMS) [.001]			
NO JUMPER	OJ					
FUSE	F		RESISTANCE (OHMS) [<10]	[5%]	RERR [1.0] RHIRESID [3.2 ohms] RLORESID [1.8 ohms]	ILIM [.128]
OPEN CONTACT	NO					
CLOSED CONTACT	NC		RESISTANCE (OHMS) [<1]			
POTENTIOMETER	RV		RESISTANCE (OHMS)	[10%]	R23 [0.5 VALUE] RERR [1.0] RHIRESID [3.2 ohms] RLORESID [1.8 ohms] ILIM [.128] PRESET FASTATG	
TRANSISTOR (NPN)	QN		MIN. GAIN		VBE [0.6] VCE [5.0] IC [10 M] FASTATG	
TRANSISTOR (PNP)	QP		MIN. GAIN		VBE [0.6] VCE [5.0] IC [10 M] FASTATG	

Notes: 1. Items enclosed in brackets ([ ]) are defaults for unspecified items.
2. See discussions in section 6.12 for program format details.
3. See discussions in section 6.12.6 for explanations of how to use flagspecs.

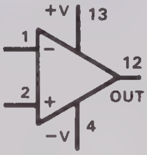
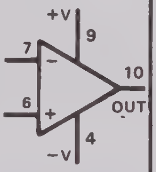
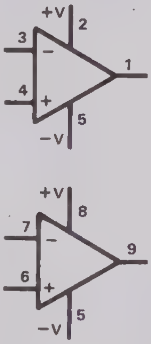
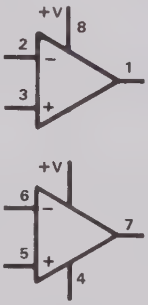
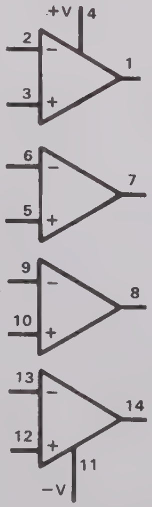
*Depends on KNEE VOLTAGE.

Table 6-4 (continued)
CIRCUIT DESCRIPTION PROGRAM ELEMENTS

DEVICE TYPE	CDL SYMBOL	PIN ASSIGNMENT	VALUE SECTION SPECIFICATIONS		
			NOMINAL CHARACTERISTIC VALUE	TOLERANCE	IMPLEMENTED TEST LIBRARY OPTIONAL FLAGSPECS
INTEGRATED CIRCUIT	(type)				KNEE [0.66 V] NOFAST CLKFREQ
INTEGRATED CIRCUIT OR DEVICE NOT RECOGNIZED BY ATG LIBRARIES	ICnn IC[40]				KNEE [0.66 V] NOFAST CLKFREQ
	ICAnn				KNEE [0.66 V] NOFAST CLKFREQ
	DICnn				
	UKnn				
USER SPECIFIED GUARD	GD				
PC BOARD	BOARD				SHORT RPT FASTATG
ANY COMPONENT					NOTEST
SUPPLIED COMPONENT LIBRARY CIRCUIT ELEMENTS					
FIELD EFFECT TRANSISTOR N JUNCTION	NJFET				
FIELD EFFECT TRANSISTOR P JUNCTION	PJFET				
SILICON CONTROLLED RECTIFIER	SCR				VA [4.0] ILIM [20 mA] IGT [2 mA]
OPAMP	SINDIP14 (SINGLE 14-PIN DIP)				
	SINFLP10 (SINGLE 10-PIN FLATPAK)				VDIFF (100 mV) POSRAIL* NEGRAIL*
	SINDIP8 (SINGLE 8-PIN DIP) SINMCP8 (SINGLE 8-PIN METAL CAN)				

Notes: 1. Items enclosed in brackets ([ ]) are defaults for unspecified items.
2. See discussions in section 6.12 for program format details.
3. See discussions in section 6.12.3 for explanations of how to use flagspecs.
4. nn in ICnn, ICAnn, and UKnn is any integer from 2 through 480.
5. nn in DICnn is 8 or an even integer from 14 through 40.
*These flagspecs are not optional. They must be specified.

Table 6-4 (continued)

DEVICE TYPE	CDL SYMBOL	PIN ASSIGNMENT	VALUE SECTION SPECIFICATIONS: NOMINAL CHARACTERISTIC VALUE	VALUE SECTION SPECIFICATIONS: TOLERANCE	VALUE SECTION SPECIFICATIONS: IMPLEMENTED TEST LIBRARY OPTIONAL FLAGSPECS
OPAMP COMPONENT LIBRARY (OPAMPLIB.ACL) ELEMENTS					
OPAMP	DUADIP14 (DUAL 14-PIN DIP)	 +V 13 1 – 12 2 + OUT –V 4			VDIFF (100 mV) POSRAIL* NEGRAIL*
	DUAFLP14 (DUAL 14-PIN FLATPAK)	 +V 9 7 – 10 6 + OUT 4 –V			
	DUAMCP10 (DUAL 10-PIN METAL CAN)	 +V 2 3 – 1 4 + 5 –V +V 8 7 – 9 6 + 5 –V			
	DUADIP8 (DUAL 8-PIN DIP)	 +V 8 2 – 1 3 + +V 6 – 7 5 + 4			
	QUADIP14 (QUAD 14-PIN DIP)	 +V 4 2 – 1 3 + 6 – 7 5 + 9 – 8 10 + 13 – 14 12 + 11 –V			

*These flagspecs are not optional. They must be specified.

3. The whole board. Entering the keyword BOARD in the component field makes the flagspecs in the value field apply to the whole board. For example,

BOARD=FASTATG;

will cause ATG to use the fast method of finding a test. With BOARD assigned to the component field, any of the following 3 flagspecs can be used in the value field.

SHORT	(threshold resistance)
RPT	(adequate test factor)
FASTATG	(fast ATG)

Refer to para 6.12.6 for a description of all flagspecs.

B. Value Field

The value field may have any of the following types of parameters:

nominal-value, tolerance, flagspec(s)

All items in the value field must be separated by commas. Also, all items in this field must apply to all components defined in the component field.

1. Nominal Value

This parameter defines the nominal characteristic value of the component(s) in the component field. The format is a dimensionless floating-point number with an optional scale factor. For example,

560, 65K, 0.1U, etc

scale-factor

NOTE

No space is allowed between the number and the scale factor.

Any of the following scale factors can be used:

MEG = 10 EXP 6 (meg)
K = 10 EXP 3 (kilo)
M = 10 EXP -3 (milli)
U = 10 EXP -6 (micro)
N = 10 EXP -9 (nano)
P = 10 EXP -12 (pico)

where 10 EXP n means 10 raised to the nth power.

As indicated in Table 6-3, some components require that a nominal value be specified. For other components, a default value is assumed, except for ICs which do not have characteristic values. ATG checks for the nominal value to be within a usual range. If it is specified outside this range, a warning will be written in the Cross Reference Report. This report and a list of unusual values are described in Section 7.

2. Tolerance

The tolerance specifies allowable variations about the nominal value. Do not specify tolerance unless a nominal value is given. Percentage sign (%) must be included. A single entry with or without plus or minus sign (e.g., -10%, +10% or 10%) will assign symmetrical tolerances about the nominal value (e.g., ±10%). Non-symmetrical tolerances must be specified with 2 entries (e.g., +10%, -5%).

3. Flagspecs

Flagspecs specify directives to ATG. These directives may affect the actions that ATG takes in selecting a test or may cause a specified message to be written when a test fails. For example,

```
C=CERR=0.5;
```

will affect the test limits on all capacitor tests;

```
C33 = 10U, 20%, MSG='CHECK INSERTION';
```

will cause the message "CHECK INSERTION" to be displayed if C33 fails during testing.

All flagspecs are recognized by an ATG library or by the ATG task. Users may add their own flagspecs to a circuit description file if they make corresponding modifications to a User library routine.

Depending on the type of entry (or entries) appearing in the component field of a value statement, flagspec(s) in the value field can characterize any of the following 5 different levels of components:

1. All components on the board (if BOARD is specified in the component field)

2. ACL (Analog Component Library) component types such as RPAC. Includes all ACL components that have been defined in the %CIRCUIT section as being of that type.

3. Individual ACL component, such as RPAC1, RPAC2, etc.. These flagspecs also apply to the primitive devices within the boundaries of the ACL component.

4. Primitive device types, such as R, C, etc. These flagspecs also apply to the ACL subcircuits in which these primitive devices are listed.

5. Individual devices such as R15, C35, etc. These flagspecs apply only to the devices listed and override any other higher-level conflicting flagspec.

When conflicts arise in flagspec assignments, lower-level flagspecs (those associated with more specific devices) have precedence over higher-level flagspecs (those associated with more general device types). For example, if the circuit description has both of the following VALUE statements:

```
R   = RERR=1.5;
R17 = 10K, 10%, RERR=1.2;
```

flagspec RERR will be 1.5 for all resistors except R17 (where RERR will be 1.2).

C. Examples

1. `R13,R26,R28 = 1.2K,+10%,-5%, RERR=1.5;`

Meaning: Resistors R13, R26 and R28 have a nominal value of 1.2 kilohms and a tolerance of +10% and -5%. The flagspec RERR sets the acceptable test limits to +25% and -12.5% (TOL + [RERR x TOL]) above and below the nominal value. The smaller tolerance figure is used in the calculations when the tolerances are asymmetrical.

2. `C33 = 10U,20%,MSG='CHECK INSERTION';`

Meaning: Capacitor C33 has a nominal value of 10 microfarads with a tolerance of 20%. The ATG process is instructed to insert the message CHECK INSERTION into the test program. This message will then be displayed when a failure is detected at C33.

3. `C = CERR=0.5;`

Meaning: The test limit for all capacitors is to be set at the component tolerance plus 0.5 times the component tolerance.

4. `BOARD = SHORT=20;`

Meaning: The threshold resistance used by ATG in all analog-continuity and low-resistance measurements is to be 20 ohms. Digital testing remains at 40 ± 30 ohms.

5. `U3 = MSG='PART NO. A123B45X';`

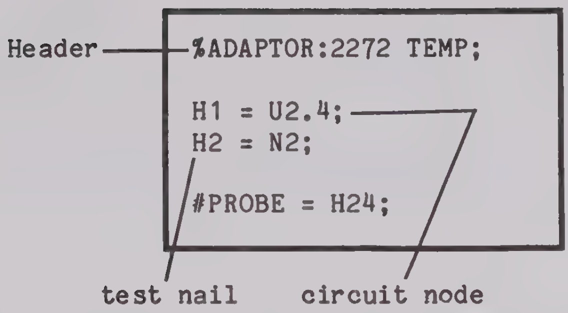
Meaning: The part number data will be copied from the circuit description file to the digital data file (.IDD) so that test results can include this data. This is a special case because ICs would not appear in the VALUE section except for the purpose of passing messages or specifying knee voltage between pins.

6. `R = FASTATG;`

Meaning: ATG uses the fast method of finding all resistance tests. The flagspec tells ATG to analyze only a limited number of the resistance test configurations in the library.

6.12.5 %ADAPTOR Section

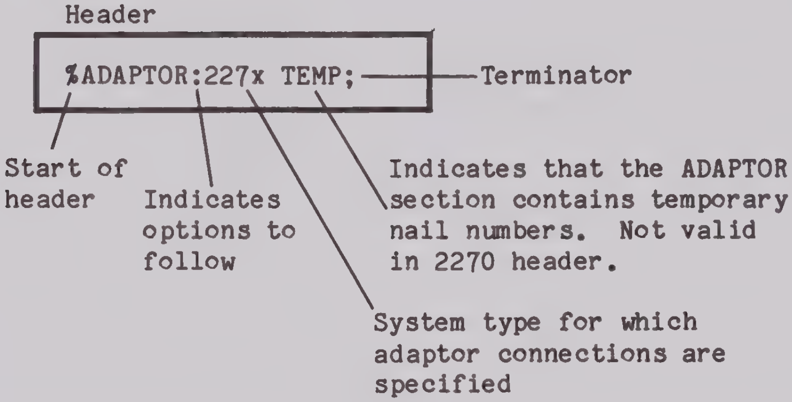
The following example illustrates the format and content of the %ADAPTOR section. Additional examples appear at the end of this paragraph.



```
%ADAPTOR:2272 TEMP;

H1 = U2.4;
H2 = N2;

#PROBE = H24;
```

Header
test nail
circuit node

The %ADAPTOR section must start with the following header statement:

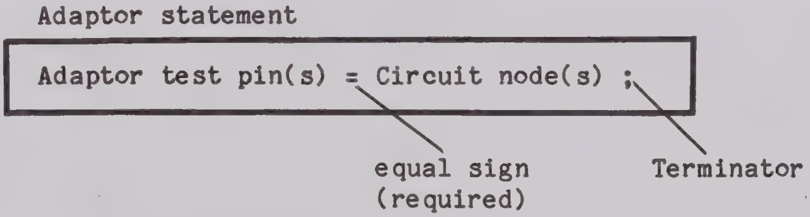


```
%ADAPTOR:227x TEMP;
```

Header
Terminator
Start of header
Indicates options to follow
Indicates that the ADAPTOR section contains temporary nail numbers. Not valid in 2270 header.
System type for which adaptor connections are specified

Note: When running ATG, if the "TEMPORARY Nail State" option is selected on the monitor page, ATG will use the %ADAPTOR:227x TEMP section if one is available, and if one is not found, it will consider an %ADAPTOR:2270 section as the one containing temporary nails. The header %ADAPTOR:2270 TEMP, however, is invalid since 2270 systems do not require temporary nails.

The adaptor statements that follow this header have 2 distinct fields, one on each side of the equal (=) sign.



```
Adaptor test pin(s) = Circuit node(s) ;
```

Adaptor statement
equal sign (required)
Terminator

The adaptor field specifies one or more test nails; the node field specifies one or more circuit nodes to which the nails are connected. Following are some general guidelines pertaining to the writing of ADAPTOR statements:

- All nodes must be assigned an adaptor-pin (test nail) in order for ATG to generate a test for the components at that node.

- Power and ground nodes should be connected for analog measurements (A or H nails) so that the analog shorts test will be performed at these nodes. Impedances on these nodes are typically too low for the digital shorts test.

- Since the .AXR report lists nodes in the order presented in the ADAPTOR section, it is recommended that nails be listed in numerical order for easy reference in the .AXR.

An explanation of the formatting and usage requirements for adaptor and node fields follows.

A. Nail Field

- Test nails are identified by a letter followed by a number (e.g., A10, H116, CD3, etc). The letter identifies the type of nail (see Table 6-5); the number specifies a particular nail within that nail type.

Table 6-5
NAIL TYPES

	Symbol	Nail Type	
	H	Hybrid	
2270 only	A	Analog-only	
	D	Digital-only	
Speedplus Option Only	F	Fast	should not be used in 2270 %ADAPTOR section
	CD	Clock Drive	
	CS	Clock Sync	
	T	Trigger	

A description of these different nail types, along with the range of numbers allowed for each type can be found in Section 8 - Nail Assignment.

- If multiple nails are specified they must be separated by either a space or comma.

- The user normally assigns nail numbers to the circuit nodes by annotating the circuit schematic in preparation to writing the circuit description.

- When preparing a circuit description for a multiplexed system (2271, 2272 or 2275), the user will typically insert temporary nail numbers in this field and indicate this fact to the system by labeling the section with the header %ADAPTOR:227x TEMP. After the system nail assignment software determines the real nail numbers, it will produce a new %ADAPTOR section labeled %ADAPTOR: 227x (i.e., 2271, 2272 or 2275).

B. Node Field

- Circuit nodes in this field are identified the same as they would be in the CIRCUIT statements; that is, either by component pin:

R51.1, U3.5, etc.

by user-assigned node name:

N1, N2, etc.

or by external name:

VCC, GND, etc.

- Test nails dedicated to the SYNC pulse or PROBE connectors on the front panel are specified by using the keywords SYNC or PROBE (preceded by a # sign) in the adaptor field. For example,

```
#PROBE = H116;
 #SYNC = H264;
```

- A circuit node can be identified by specifying a single component pin, node name or external name. Additional entries are simply cross-checked with the connection data in the CIRCUIT statements.

C. Examples

1. A15 = C33.N;

Meaning: Test nail 15 is an analog nail connected to the circuit node that contains C33 pin N.

2. D9 = U2.14;

Meaning: Test nail 9 is a digital nail connected to circuit node that contains U2 pin 14.

3. A3,D115 = U2.4;

Meaning: Analog test nail 3 and digital test nail 115 are both connected to the same node as U2 pin 4.

4. H6 = IC6.2 R3.1;

Meaning: Test nail 6 is a hybrid nail used for both analog (R3.1) and digital (IC6.2) measurements.

5. #PROBE = H255;

Meaning: Adaptor pin 255 is a hybrid nail dedicated for use by the diagnostic probe.

6.12.6 Flagspec Descriptions

The following flagspec descriptions are presented in groups according to the characteristic tests for which they apply.

A. Universally Used Flagspecs

FASTATG
MSG
NOTEST
RPT
NOCOM
SHORT

FASTATG
Recognized in several library routines. Causes ATG to speed up the search for an analog test by trying only those test configurations that are most likely to be acceptable. This frequently means that the test selected may not be the most accurate, but ATG will run much quicker.

The routines in which FASTATG is usable are resistance, capacitance, inductance, potentiometer, and transistor tests.

MSG

Tells ATG to insert a character string into the test program at the applicable procedure(s). The entry

```
CR = MSG = 'Check insertion';
```

would cause the "check insertion" to appear at all diode tests.

Messages may have up to 72 characters. All characters are allowed except an apostrophe (')*, a single %* or a CR-LF control code combination indicating a new line. A new line instruction must be inserted at the end of the message inside the tick mark or the next printed item will be on the same line. The new line instruction is "%%NL%%". The message may be assigned to any component. The message that appears on the strip printer does not stand out. Compensation for this may be achieved by adding leading and trailing special characters, such as an asterisk, to the message. Another method of highlighting the message would be to double the character width in the message. This is accomplished by starting the message with "%%036%%". End the message with %%035%% to return to normal width. Examples of these two methods are:

```
MSG = '****CHECK INSERTION****%%NL%%'

MSG = '%%036%% CHECK INSERTION%%035%%%%NL%%'
```

NOTEST

Tells ATG not to look for a test (analog or digital) for this component. The user may know that there is not an adequate test for the component and can save ATG the time it would take to look for a test and print a message that says a test could not be found in any library.

RPT

Allows the user to set the error limits used by ATG to classify a test as a "superior" test. A superior test is one whose measurement error divided by the component tolerance is less than the RPT. The default value is 1, which means the error could be equal to a value within 100% of the tolerance value (or some other value determined by the library procedure) and be classified as "superior." This flagspec is recognized by ATG at the time that the report statistics are compiled, and applies to all components that have measurement error specifications.

NOCOM

This flagspec added to the .CKT allows appropriate form tests to be written into the .TPG and compiled.

---

* A % can be included in a message by using 4 % symbols (%%%%) where the % symbol is desired. ATG will strip off the first and last %s as special code delimiters and will pass %% to the translator. The translator will strip off the first of these 2, leaving a % in the run time message. The apostrophe can be included in a message by enclosing it in dual percents (%%'%%). ATG and the translator will each strip off a pair of %s.

SHORT

This flagspec allows the user to change the resistance limit below which a component is defined as shorted. (The default is 10 ohms.)

B. Resistance Test Flagspecs

RERR
RHIRESID
RLORESID
ILIM
DCR
RTOL
RLOV
RHIV

RERR

Used to set error limits that determine if ATG accepts the test which it just evaluated and stops looking for a better test. The values that are specified are treated as an add-on factor to the component tolerance, i.e., (TOL + [RERR x TOL]). When RERR=1 (the default value), the test limits are equal to twice the tolerance. For example, if the nominal value is 1K and the tolerance is ±10%, the test limits would be within ±20% of the nominal value or between 800 and 1200 ohms. Similarly, if RERR is specified as 0.85 and tolerance 10%, the test limits would be 18.5% of the nominal value. See Para 8.2.3 in the 227X Test Library Programming Manual for more information on RERR.

RHIRESID
RLORESID

There is a certain amount of residual resistance in the system due to mux, scanner, and fixture cabling. This resistance can introduce a significant error when the component being tested has a small resistance value. It is also evident that this residual resistance will vary from one connection to the next. ATG includes a scheme that compensates for this variable resistance. The user can determine a high and a low value of residual resistance for each applicable component. These measured values are inserted into the circuit description %VALUE section as flagspecs, named RHIRESID and RLORESID. The default values for these are RHIRESID=3.2 ohms and RLORESID=1.8 ohms. ATG calculates a nominal value of residual resistance and adds it to the component test limits. It treats the variation in residual resistance as measurement error.

ILIM

Used to specify the limiting value of source current allowable during a test. The default value is 0.128 A. The only other value that can be selected with the standard voltage source is 0.020 A.

DCR

This flagspec specifies the dc resistance value of an inductor. There is a default value of 1.0 ohm; however, this default value is only for determining circuit impedances for other tests. If a dc resistance test on an inductor is desired, both DCR and RTOL must be specified.

RTOL

This flagspec specifies the tolerance of the dc resistance of an inductor. There is no default for this flagspec. It must be specified for an inductor or no resistance test will be generated.

RLOV

This flagspec is used to save ATG processing time for devices with low resistance values. ATG will bypass all library test configurations that would use a source voltage greater than 0.2 volt when the device resistance is less than or equal to the value specified in this flagspec. The default value is 500 ohms.

RHIV

This flagspec is used to provide a more accurate test of devices with a high resistance. ATG will go directly to a guarded test configuration that uses a 5-volt source when the component resistance is greater than the value specified in this flagspec. The default value is 1.0 megohm.

C. Potentiometer Test Flagspecs

R23
PRESET
All resistance test flagspecs except RTOL and DCR

R23

Resistance between the wiper (pin 2) and the clockwise end of the pot (pin 3). Normal pot convention has the wiper move from pin 1 to pin 3 when turned clockwise. Therefore, this flagspec will always cause the resistance between pin 1 and the wiper to increase when the pot is adjusted in the clockwise direction.

PRESET

Keyword indicating to ATG that the pot is pre-set to a known setting.

Depending on what flagspecs are specified, the following pot tests are performed:

1. No flagspecs defined   For example,

   RV7 = 10K, 20%;

   In this case, the resulting test program will guide the user (via messages on the display) in adjusting the pot to its mid-point value if the pot is used as a divider. If the pot is a variable resistor (wiper tied to one end), the displayed messages will guide the user to adjust the pot to its maximum value.

2. Flagspec R23 defined and PRESET defined   For example,

   RV7 = 10K, 20%, R23=1K, PRESET;

   This causes ATG to test the pot as two fixed resistors (e.g., 1K and 9K). The pot must be preset to the specified setting.

3. Flagspecs R23 defined (Adj.)   For example,

   RV7 = 10K, 20%, R23=1K;

   This will cause the resulting test program to guide the user in adjusting the pot to the value specified by R23.

D. Capacitance Test Flagspecs

CERR	CHIRESID
FREQ	CLORESID

CERR
This flagspec is used in the same manner as RERR. The default is 1.0.

FREQ
Selects the source test frequency for capacitors from 100 Hz, 1 kHz, or 10 kHz. The default frequency is optimally determined by the test system.

CHIRESID
CLORESID
These flagspecs are used in the same manner as the resistance residual flagspecs. The default values are CHIRESID = 20 pF and CLORESID = 5 pF.

E. Inductance Test Flagspecs

LERR	LHIRESID
FREQ	LLORESID

LERR
This flagspec is used in the same manner as RERR. The default is 1.0.

FREQ
Selects the source test frequency. If it is not specified in the circuit description, the ATG chooses a frequency that optimizes for minimum instrument error and is based on the expected value of inductance. This is done according to the following list:

Inductance Range	Source Frequency
Less than 5.1 mH	10 kHz
5.1mH to 2.6 H	1 kHz
More than 2.6 H	100 Hz

LHIRESID
LLORESID
These flagspecs are used in the same manner as the resistance residual flagspecs. The default values are LHIRESID = 15 uH and LLORESID = 5 uH.

F. Transistor Test Flagspecs

VBE
VCE
IC

VBE
Base-emitter voltage for the transistor under test. The default value is 0.6 volt.

VCE
Collector-emitter voltage for the transistor under test. The default value is 5.0 volts.

IC

Collector current. Useful for specifying higher current values when testing high-power transistors. The maximum source current available is 100 mA. This will be shared by the transistor and shunt circuitry. The default value is 10 mA.

G. Diode Test Flagspecs

DR	IFV
DRH	IDR
DRL	FVONLY

DR

Nominal expected value of dynamic resistance of the diode. The default is 5.5 ohms when the forward voltage is greater than 0.5 volt; otherwise, it is 20 ohms. Used in forward voltage and dynamic resistance tests.

DRH

Maximum value of dynamic resistance. The default is 10 ohms when DR=5.5 and 30 ohms when DR=20. Otherwise, it is 1.5 times DR. If DRH is specified less than 1.2 times DR, ATG will override and use the default.

DRL

Minimum value of dynamic resistance. The default is 0 ohm when DR=5.5 and 8 ohms when DR=20. Otherwise, it is 0.5 times DR. If DRL is specified greater than 0.8 times DR, ATG will override and use the default.

IFV

Forward voltage test current. The default is 20 mA. If it is specified less than 10 mA, ATG will override and set it to 10 mA. If it is specified greater than 40 mA, ATG will override and set it to 40 mA.

IDR

Dynamic resistance test current. The default is 5 mA. If it is specified less than 5 mA, ATG overrides and sets to 5 mA. If it is specified greater than 20 mA, ATG will override and set to 20 mA. If IFV is not greater than IDR by at least 5 mA, ATG will override and set IDR to IFV -5 mA.

FVONLY

The forward knee voltage/dynamic resistance tests are evaluated but the dynamic resistance test is not written.

H. Zener Diode Test Flagspecs

IZT ZZT

IZT

Zener test current which defaults to a value determined by dividing the number 124.8 by VZ.

ZZT

Zener test resistance which defaults to a value determined by multiplying the number 1.06 by VZ.

I. SCR Test Flagspecs

VA   ILIM   IGT

VA

Anode voltage for the SCR under test. The default value is 4.0 volts.

ILIM

Current limit for the voltage source. The only values that can be selected are 0.02 A and 0.128 A. The default value is 0.02 A.

IGT

Gate current used to trigger the SCR. The default value is 2 mA. If a value less than 0.1 mA or greater than 10 mA is specified, ATG will override and use 2 mA.

J. Op Amp Test Flagspecs

POSRAIL   NEGRAIL   VDIFF

POSRAIL

Normally the positive supply voltage. If only a negative supply is used, set to zero. Results in a minimum (lowest) acceptable positive voltage, referred to ground, that the output must reach when the amplifier is driven positive. The actual test limit is POSRAIL - 10%. This flagspec must be specified if the analog component is an op amp. No default is provided.

NEGRAIL

Normally the negative supply voltage. If only a positive supply is used, set to zero. Results in maximum (highest) acceptable negative voltage, referred to ground, that the output must reach when the amplifier is driven negative. The actual test limit is NEGRAIL + 10%. This flagspec must be specified if the analog component is an op amp. No default is provided.

VDIFF

Differential input voltage. This is an optional flagspec. ATG provides a default value of 100 mV.

K. Digital Flagspecs (SpeedPlus Option)

NOFAST   CLKFREQ

NOFAST

Prevents digital ATG from writing any fast tests for the component (or components) within the scope of this flagspec. If a slow test exists, it is chosen; if no slow test exists - no test is selected.

CLKFREQ

Specifies the frequency in Hz of the DUT (device-under-test) clock.

ATG Procedures – Section 7

7.1 INTRODUCTION

7.1.1. ATG Process

Automatic Test Generation (ATG) consists of the 6 processes that are initiated when the appropriate options are selected on the ATG Monitor Options page. These processes are represented in Figure 7-1 by a box that shows the monitor page option name; the various input and output files for each process are also shown. Four activities are shown that would only be performed as a result of errors and unexpected results. The flow to, and from, these activities is represented by dashed lines.

Tables 7-1 and 7-2 define the function of each process and file used by ATG. Paragraphs 7.2 through 7.7 explain the processes.

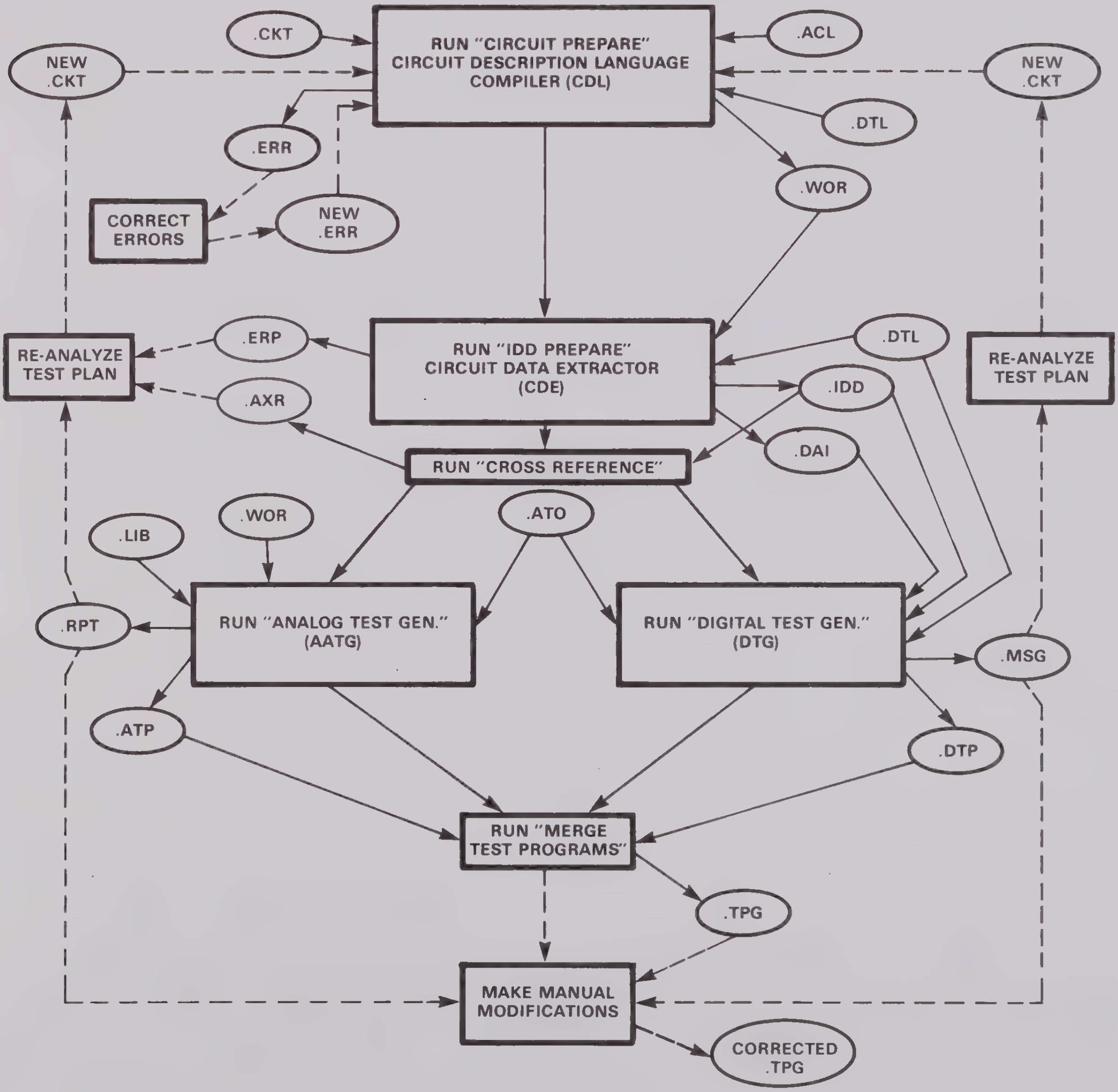

.CKT
RUN "CIRCUIT PREPARE" CIRCUIT DESCRIPTION LANGUAGE COMPILER (CDL)
.ACL
NEW .CKT
NEW .CKT
.DTL
.ERR
.WOR
CORRECT ERRORS
NEW .ERR
RE-ANALYZE TEST PLAN
.ERP
RUN "IDD PREPARE" CIRCUIT DATA EXTRACTOR (CDE)
.DTL
RE-ANALYZE TEST PLAN
.AXR
.IDD
.DAI
RUN "CROSS REFERENCE"
.WOR
.ATO
.LIB
RUN "ANALOG TEST GEN." (AATG)
RUN "DIGITAL TEST GEN." (DTG)
.RPT
.MSG
.ATP
.DTP
RUN "MERGE TEST PROGRAMS"
.TPG
MAKE MANUAL MODIFICATIONS
CORRECTED .TPG

Figure 7-1. ATG Process.

Table 7-1
DEFINITION OF ATG PROCESSES

Name	Function	Option
Circuit Description Language Compiler (CDL)	Compile a machine-readable work file (.WOR) from the Circuit Description file (.CKT). Include applicable entries from the Analog Components Library files (.ACL) and component information from the .DTL files.	Circuit Prepare
Circuit Description Data Extractor (CDE)	Generate a Digital ATG Interconnect file (.DAI) and an In-circuit Diagnostic Data file (.IDD or .IDX) from the .WOR and .DTL files.	IDD Prepare
Cross Reference Report Generator	Prepare Cross Reference Report (.AXR) which summarizes board connection data.	Cross Reference
Analog Test Generator (AATG)	Generate an Analog Test Program file (.ATP) from the .WOR, .LIB's, and .ATO or .ATX files.	Analog Test Gen.
Digital Test Generator (DTG)	Generate a Digital Test Program file (.DTP) from the .DAI, .IDD or .IDX, .DTL's and .ATO or .ATX files.	Digital Test Gen.
Merge	Combine the .ATP and .DTP files to make a source Test Program (.TPG or .TPX) files.	Merge Test Programs

Table 7-2
DEFINITION OF ATG FILES

Extension	Name	Purpose
.CKT	Circuit Description	Source file created by the user to describe the pc board circuit to the test system.
.ACL	Analog Components Library	A library file that contains descriptions of complex devices.
.WOR	Circuit Description Work File	A circuit description file that has been compiled into a machine-usable format.
.ERR	Circuit Description Error File	A file containing CDL compile error messages.
.DTL	Digital Test Library	A library file that contains test routines for digital devices.

Table 7-2 (continued)

Extention	Name	Purpose
.IDD or .IDX	In-circuit Diagnostic Data File	Contains Interconnect information for use by DTG and Run-time diagnostics.
.DAI	Digital ATG Interconnect File	Contains Driver/Sensor and bus connection information for use by .DTG.
.AXR	Analog Cross Reference Report	Contains connection lists by components and by nodes and reports discrepancies.
.ERP	Extraction Report	Contains summary of device node and nail entries in .DAI and .IDD files.
.ATO or .ATX	Automatic Test Options File	User-created file of test program statements, routines, and instructions to ATG for options in test generation.
.LIB	Analog Test Library	Contains selection procedures for generating basic analog component tests.
.ATP	Analog Test Program	Program generated by AATG for testing analog components
.DTP	Digital Test Program	Program generated by DTG for testing digital components.
.RPT	Analog ATG Report File	Contains result data for all components tested, showing libraries used and any problems with generating tests.
.MSG	Digital ATG Message File	Contains result data for all components tested, showing libraries used and any problems with generating tests.
.TPG or .TPX	Test Program	Source test program that includes analog and digital test procedures.
.FTP	Fixture Test Program	Program generated to check that each nail in a fixture makes proper contact to a plate mounted on the fixture.
.CTP	Continuity Test Program	Program generated to check opens and shorts on a bare board.

7.1.2 Customer Training Board (CTB)

The pc board available to users for training purposes was introduced in Section 4. It is used in this section to illustrate ATG processes and results. The schematic is shown, with final nail numbers, in Figure 6-22. The CTB is used to illustrate each file that the ATG creates. These files are presented in the remaining paragraphs of this section. Detailed comments and explanations accompany these listings. The board name used for the .CKT and other CTB-related files is "TRAINGBD."

7.2 RUN CIRCUIT DESCRIPTION COMPILER

The circuit description file (.CKT) created in accordance with Section 6, is compiled by selecting YES or EDIT_ON_ERRORS for the "Circuit prepare" option on the ATG monitor options page. (Refer to para 7.9 for a complete description of the ATG options page.) The compiled file is the circuit description work (.WOR) file that is used by ATG in the test generation process. The compile is done in 3 passes of the .CKT file. The first pass includes a syntax check of all the .CKT statements; the second pass, a check for interconnection discrepancies, and the third pass checks for SpeedPlus pin use discrepancies. A list of all error messages, that may occur during the CDL process, are listed in Appendix B.

The basic process for compiling and debugging a .CKT file is shown in Figure 7-2. Figure 7-3 is a more detailed diagram showing the effects of the various options. When errors occur, either the original .CKT file may be edited or a full listing .ERR file may be edited. The .ERR file is a full listing file if the "Format of report" option has FULL selected or if the "Circuit prepare" option has EDIT_ON_ERRORS selected. The error messages that appear in the .ERR file are ignored by the compiler when the next attempt to compile is made. Refer to Section 11 of this manual for instructions on editing.

The editor is automatically invoked after a pass that contained errors, if the EDIT_ON_ERRORS option is selected on the ATG page. In this case, the pass will be automatically re-run when the editor is exited. When an .ERR file has been corrected and successfully compiled, that file automatically is given the .CKT extension so that future use of the .CKT is with the corrected version.

In order to get out of the EDIT_ON_ERRORS loop prior to obtaining an error-free .CKT, enter one of the following edit commands when in the editor.

ESC E	Use this command when running from the GR terminal, which invokes the TECO editor.
PF1 CTRL E	Use this command when running from the VT10X terminal, which invokes the ED2 editor.

As CDL is running, certain messages appear on the display. First

```
BEGIN CDL COMPILE PASS 1
```

appears. if there are no errors or warnings, the next message,

```
BEGIN CDL COMPILE PASS 2
```

appears, and so on for pass 3. When the compile is completed with no errors, the message

```
COMPILATION COMPLETED, 0 ERROR(S) DETECTED
```

If any errors or warnings are detected, error messages are displayed for each and the compile stops at the end of the pass where they are detected. The .ERR file also contains these messages. The error message format contains: (1) the line of data containing the error, (2) an underlining caret (^) pointing at the location of the error, (3) the error number, starting at 1, indicating how many errors in this pass, (4) the number of the line in the source file containing this error, and (5) CDL1, CDL2, or CDL3,

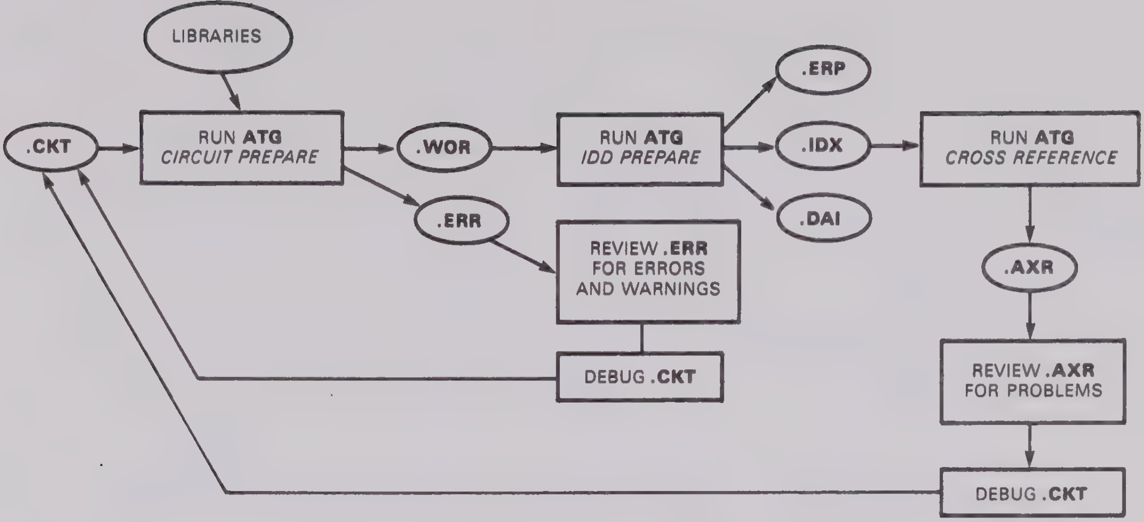

LIBRARIES
.CKT
RUN ATG
CIRCUIT PREPARE
.WOR
.ERR
RUN ATG
IDD PREPARE
.ERP
.IDX
.DAI
RUN ATG
CROSS REFERENCE
REVIEW .ERR
FOR ERRORS
AND WARNINGS
DEBUG .CKT
.AXR
REVIEW .AXR
FOR PROBLEMS
DEBUG .CKT

Figure 7-2. Compiling and Debugging a .CKT.

indicating whether pass 1, pass 2, or pass 3 is being run, followed by a descriptive error message. The following is an example of what appears on the display when errors are detected in pass 1.

Example

```
BEGIN CDL COMPILE PASS 1

CR1  D  A  =N22,  C  =N15;
     ^

### WARNING No. 1 AT LINE ?:
CDL1 - Device not in any library, defaults to 40-pin IC.

CR4  CR  A  =N22,  C  =9.3  CR4.  CR3.C;
                              ^

### ERROR No. 1 AT LINE X:
CDL1 - Pin name must be given explicitly.

2 ERROR(S) DETECTED
TERMINATED AFTER PASS 1
```

Meaning: The wrong device type was entered for the component CR1. D was entered instead of CR. Type D did not exist in a library as a component type. When a device type is not found, ATG will assume a 40-pin IC. Therefore, a warning is indicated rather than an error. The message indicates it was the first warning found and it occurred at line ? of the .CKT. The first error resulted from the pin name of component CR4 in the connection list of line X being missing.

Pass 2 begins as soon as pass 1 is successfully completed. Pass 2 checks the connection data. Once a connection is defined, the compiler uses subsequent references to that connection to cross-check the original statement. If detectable connection errors are found, an error message is inserted in the .ERR file and is also displayed on the screen. The following is a typical example.

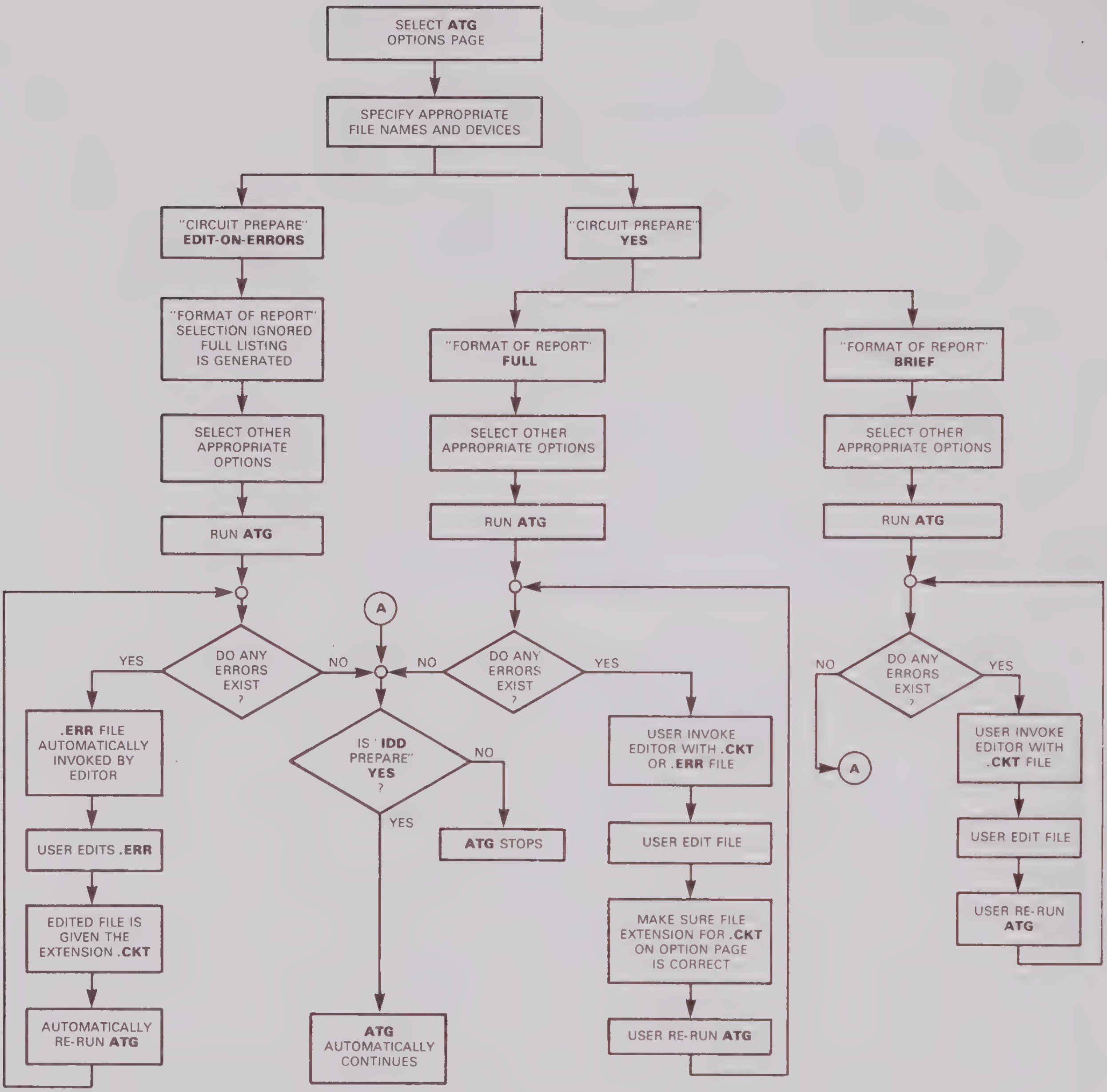

SELECT ATG OPTIONS PAGE
SPECIFY APPROPRIATE FILE NAMES AND DEVICES
"CIRCUIT PREPARE" EDIT-ON-ERRORS
"CIRCUIT PREPARE" YES
"FORMAT OF REPORT" SELECTION IGNORED FULL LISTING IS GENERATED
"FORMAT OF REPORT" FULL
"FORMAT OF REPORT" BRIEF
SELECT OTHER APPROPRIATE OPTIONS
SELECT OTHER APPROPRIATE OPTIONS
SELECT OTHER APPROPRIATE OPTIONS
RUN ATG
RUN ATG
RUN ATG
A
YES
DO ANY ERRORS EXIST ?
NO
NO
DO ANY ERRORS EXIST ?
YES
NO
DO ANY ERRORS EXIST ?
YES
.ERR FILE AUTOMATICALLY INVOKED BY EDITOR
IS "IDD PREPARE" YES ?
NO
USER INVOKE EDITOR WITH .CKT OR .ERR FILE
A
USER INVOKE EDITOR WITH .CKT FILE
YES
USER EDITS .ERR
ATG STOPS
USER EDIT FILE
USER EDIT FILE
EDITED FILE IS GIVEN THE EXTENSION .CKT
MAKE SURE FILE EXTENSION FOR .CKT ON OPTION PAGE IS CORRECT
USER RE-RUN ATG
AUTOMATICALLY RE-RUN ATG
ATG AUTOMATICALLY CONTINUES
USER RE-RUN ATG

Figure 7-3. Effect of Circuit Prepare Options.

Example

```
BEGIN CDL COMPILE PASS 2

R1  R  N16,  N15;
         ^

### ERROR No. 1 AT LINE 19:
CDL2 - This pin connected to node at line nn, device pin at line
nn.

H19  =  N119;
           ^

### ERROR No. 2 AT LINE 24:
CDL2 - This device or pin name has not been defined anywhere.
```

```
2 ERROR(S) DETECTED
TERMINATED AFTER PASS 2
```

Meaning: The node names are reversed for the pins of R1 and the compiler determined that the connections do not agree with those defined in the connection statement for C1. An error in typing the node name to which nail 19 is connected. No such node was defined in the connection section.

The user again makes appropriate corrections to the source file and reruns ATG until pass 2 is successfully completed. Then pass 3 is run in a similar way to check the optional SpeedPlus connection data.

At any time during compilation, the user may request the latest compiler status by pressing the HELP key on the GR terminal, or CTRL A on the VT100 terminal. The program responds by displaying the following message type:

```
CDL Compiler Status - -
Line 24 of Pass 1 (2 errors detected)
A 15 = CR4.C, GND
```

which gives the line number and contents of the source file statement being processed, the pass number, the number of errors detected so far. Each time HELP is pressed, an updated status is displayed and the old status is moved up on the screen.

The user can abort a compilation at any time by pressing the RESET button. Compilation can also be aborted during pass 1 or pass 3 by typing CTRL-C. This causes the following message to appear on the screen.

```
OPERATOR ABORTED
4 ERRORS DETECTED
```

Warnings are printed in the .ERR file to indicate certain things that are suspect. Therefore, it is highly recommended that the file be looked at for these warnings and consideration be given to them. This could save many problems in a later process.

The completed .ERR for TRAINGBD is partially listed here to show differences from the original .CKT. Remember that when the .ERR is created with EDIT-ON-ERRORS, the extension is automatically changed to .CKT. Comments and examples of error messages are included beside the listing.

TRAINGBD.ERR

```
###     CDL FILE:  SC:TRAINGBD.CKT,       COMPILED 27-OCT-83
###
#####  WARNING NO  1 AT LINE 77:
###  CDL1- LM1458 not found in any library, defaults to type IC.
###
#####  WARNING NO  2 AT LINE 78:
###  CDL1- RPAK8 not found in any library, defaults to type IC.
###
#####  WARNING NO. 3 AT LINE 112:
###  CDL1- NE5118 not found in any library, defaults to type IC.
###
#####  WARNING NO. 4 AT LINE 123.
###  CDL1- XTL not found in any library, defaults to type IC.
###
     1 :/* SC:TRAINGBD CKT Created by GenRad Circuit Generator */
     2 :
     3 :
     4 :
     5 :
     6 :
     7 :
     8 :
     9 :
    10 :
    11 :
    12
    13 :
    14 :
    15 :
    16 :
    17 :%CIRCUIT;
    18 :
    19 :
    20 :GND        EXT        ,      N16;
    21 :VCC        EXT        ,      N15;
    22 :VCC15      EXT        ,      N17;
    23 :VCC15N     EXT        ,      N18,
    24 .
    25 :C2         C          N53, N18;
    26 :C3         C          N181, N117;
    27 :C4         C          N84, N16,
    28 :C5         C          N83, N16;
    29 :C6         C          N51, N16,
    30 :C7       \ C          N52, N16;
    31
    32 :C1         CP1        P = N15, N = N309,
    33 :
    34 :CR1        CR         A = N309, C = N16,
    35 :CR2        CR         A = N79, C = N81;
    36 :CR4        CR         A = N16, C = N82;
    37 :DS1        CR         A = N80, C = N45;
    38 :
    39 :Q1         QN         B = N277, C = N115, E = N16,
    40
    41 :R1         R          N309, N277,
    42 :R12        R          N118, N16;
    43 :R13        R          N77, N16;
    44  R14        R          N77, N54;
    45 :R15        R          N21, N79;
    46 :R16        R          N18, N81,
    47 :R17        R          N85, N84,
    48 :R18        R          N17, N85,
    49 :R19        R          N45, N47,
    50 :R2         R          N277, N16,
    51 :R20        R          N86, N49;
    52 :R21        R          N15, N50,
    53 .R22        R          N50, N51,
    54 .R23        R          N22, N115,
    55  R24        R          N15, N117,
    56 .R3         R          N15, N115,
    57 :R4         R          N115, N113;
    58 :R5         R          N15, N269,
    59
    60 :/* SC.TRAINGBD CKT Created by GenRad Circuit Generator */
    61 :
    62
    63 :
    64 :R6         R          N15, N270;
    65 :R7         R          N15, N307;
    66 :R8         R          N15, N273;
    67 :R9         R          N15, N272;
    68 :
    69 :R100       RV         N54, N54, N21,
    70 :
    71 :CR3        VZ         A = N81, C = N82;
    72 :
    73 :Q3         SCR        A = N47, C = N16, G = N86;
    74 :
    75 :Q2         NJFET      G = N79, D = N80, S = N21;
    76 :
    77 :U6         LM1458     N181, N117, N118, N18, N181, N77, N21, N17;
    78 :Z1         RPAK8      N15, N302, N304, N113, N143, N207, N146, N211,
    79 :                    N210;
    80 :
```

Comments

The first line of the .ERR file identifies the .CKT file and the date on which it was compiled. The .CKT file is shown in para 6.11. Any warning messages, generated during pass 1 and pass 2 of the compile, appear next. The TRAINGBD.CKT contains 4 device types that have no test routine in any system or update library. Therefore, 4 warning messages are inserted by CDL pass 1 at the beginning of the .ERR.

The warnings alert the user of any device that cannot be tested because of no routine in any system or update library. The user can then add a test to a user library or to an .ATX file if not already done. In the case of the TRAINGBD, a user analog components library (.ACL) file exists for testing RPAK8 and the file TRAINGBD.ATX exists for testing an LM1458 and an NE5118. The complete .ATX file is printed at the end of Section 5. No test is provided for the XTL.

Following the warnings is a complete listing of the .CKT file with line numbers inserted at the beginning of each line. Any message added by CDL (heading, warnings, etc.) are preceded by 3 # symbols.

TRAINGBD.ERR (continued)

```
     81 .
     82 :
     83 :U1        7400      N301, N302, N303, N303, N304, N305, N16, N110,
     84 :                   N212, N111, N112, N110, N111, N15;
     85 :
     86 :U10       7414      N115, N114, N114, N113, N209, N111, N16, N13, N78,
     87 :                   N19, N14, N116, N20, N15;
     88 .
     89 :U11       2114      N176, N177, N178, N179, N142, N141, N180, N110,
     90 :                   N16, N109, N206, N205, N244, N243, N173, N174,
     91 :                   N175, N15;
     92 :
     93 :U2        Z80CPU    N274, N275, N276, N237, N238, N272, N243, N242,
###
###  WARNING NO. 5 AT LINE 93:
###  CDL2- Fast nail desired at pin 8 nail H242.
###
     94                     N244, N205, N15, N241, N206, N239, N240, N210,
###                                 ^
###  WARNING NO  6 AT LINE 94:
###  CDL2- Fast nail desired at pin 9 nail H244.
###
###
###  WARNING NO. 7 AT LINE 94.
###  CDL2- Fast nail desired at pin 10 nail H205.
###
###
###  WARNING NO. 8 AT LINE 94:
###  CDL2- Fast nail desired at pin 12 nail H241.
###
###
###  WARNING NO. 9 AT LINE 94
###  CDL2- Fast nail desired at pin 13 nail H206
###
###
###  WARNING NO. 10 AT LINE 94
###  CDL2- Fast nail desired at pin 14 nail H239
###
     95 :                   N211, N207, N209, N208, N144, N109, N145, N143,
     96 :                   N146, N113, N147, N148, N16, N142, N141, N180,
     97 :                   N179, N178, N177, N176, N175, N174, N173, N212,
     98 :
     99 ·


    100 :/* SC:TRAINGBD.CKT Created by GenRad Circuit Generator */
    101 :
    102 ·
    103 :
    104 :U3        2114      N176, N177, N178, N179, N142, N141, N180, N110,
    105 :                   N16, N109, N242, N241, N240, N239, N173, N174,
    106 :                   N175, N15;
    107 :
    108 :U4        2716      N175, N176, N177, N178, N179, N180, N141, N142,
    109 :                   N239, N240, N241, N16, N242, N243, N244, N205,
    110 :                   N206, N112, N212, N112, N15, N173, N174, N15,
    111 :
    112 :U5        NE5118    N16, N239, N240, N241, N242, N243, N244, N205,
    113 :                   N206, N208, N213, N245, N16, N149, N245, N53, N18,
    114 :                   N150, N17, N181, N117, N16;
    115 :
    116 :U7        NE555     N16, N84, N82, N17, N83, N84, N85, N18;
    117 :
    118 :U8        NE555     N16, N51, N49, N15, N52, N51, N50, N15;
    119 :
    120 :U9        7474      N307, N306, N305, N269, N308, N306, N16, N271,
    121 :                   N272, N270, N308, N271, N273, N15;
    122 :
    123 :Y1        XTL       7 = N16, N301, 14 = N15;
    124 :
    125 :%VALUE;
    126 :
    127 :
    128 :C2          = 10.0N,
    129 :C3          = 20.0P;
    130 :C4          = 1.0U;
    131 :C5          = 10.0N;
    132 :C6          = 10.0U;
    133 :C7          = 10.0N,
    134 :
    135 :C1          = 10.0U;
    136 :
    137 :
    138 :
    139 :R1          = 1.0K;
    140 :R12         = 5.1K;
    141 :R13         = 10.0K;
    142 :R14         = 3.3K;
    143 :R15         = 100.0K;
    144 :R16         = 4.7K;
    145 :R17         = 390.0K;
    146 .R18         = 1.0MEG;
    147 :R19         = 1.0, +1%, -1%, RLORESID=1;
    148 :R2          = 1.0K;
    149 :R20         = 3.3K;
    150 :R21         = 510.0K;
    151 :R22         = 470.0K;
    152 :R23         = 10.0K;
    153 :
```

Comments

Any warnings generated by pass 3 of the compile appear in the listing following the line which caused the warning to be generated. The TRAINGBD.CKT contains a Z80 CPU whose preferred test, in the digital library, requires FAST test points. Since no FAST test points were specified in the .CKT, a warning is generated at each pin that requires a FAST test point. The TRAINGBD.ATX file tells ATG to use the normal speed test for the Z80 instead of the fast test.

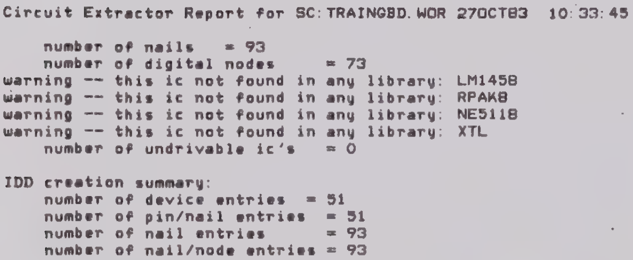
CDL stops inserting warnings after 6 have been generated in any pass. There are more than 6 FAST test points required for the Z80 but only the first 6 that were encountered appear in the .ERR.

```
 154 :/* SC:TRAINGBD.CKT Created by GenRad Circuit Generator */
 155
 156
 157
 158 :
 159 :R24         = 10.OMEG, +5%, -5%;
 160 :R3          = 10.0K,
 161 :R4          = 10.0K;
 162 :R5          = 3.3K,
 163 :R6          = 3.3K;
 164 :R7          = 3.3K;
 165 :R8          = 3.3K;
 166 :R9          = 330.0;
 167 :
 168 :R100        = 10.0K, 20%,
 169 :
 170 :CR3         =  MSG='7.5 ZENER DIODE';
 171 :
 172
 173 :U6          = POSRAIL=15, NEGRAIL=15,
 174
 175
 176 :
 177 :%ADAPTOR : 2275 TEMP,
 178
 179
 180
 181
 182 :H13    = N13;        H14    = N14;        H15    = N15,
 183 :H16    = N16;        H17    = N17;        H18    = N18;
 184 :H19    = N19;        H20    = N20;        H21    = N21;
 185 :H22    = N22;        H46    = N45;        H48    = N47,
 186 :H49    = N49;        H50    = N50;        H51    = N51;
 187 :H52    = N52;        H53    = N53;        H54    = N54,
 188 :H77    = N77;        H78    = N78;        H79    = N79,
 189 :H80    = N80;        H81    = N81;        H82    = N82,
 190 :H83    = N83;        H84    = N84,        H85    = N85,
 191 :H86    = N86;        H109   = N109;       H110   = N110;
 192 :H111   = N111,       H112   = N112;       H113   = N113,
 193 :H114   = N114;       H115   = N115;       H116   = N116,
 194  H117   = N117;       H118   = N118;       H141   = N141,
 195 :H142   = N142,       H143   = N143;       H144   = N144,
 196 :
 197 :/* SC:TRAINGBD.CKT Created by GenRad Circuit Generator */
 198 :
 199 :
 200
 201 :H145   = N145;       H146   = N146;       H147   = N147;
 202  H148   = N148;       H149   = N149;       H150   = N150;
 203 :H173   = N173;       H174   = N174;       H175   = N175;
 204  H176   = N176;       H177   = N177;       H178   = N178;
 205 :H179   = N179;       H180   = N180;       H181   = N181;
 206 :H205   = N205;       H206   = N206;       H207   = N207;
 207 :H208   = N208;       H209   = N209;       H210   = N210;
 208 :H211   = N211;       H212   = N212;       H213   = N213,
 209 :H237   = N237;       H238   = N238;       H239   = N239;
 210 :H240   = N240;       H241   = N241;       H242   = N242;
 211 :H243   = N243;       H244   = N244;       H245   = N245;
 212 :H269   = N269;       H270   = N270;       H271   = N271;
 213 :H272   = N272;       H273   = N273;       H274   = N274;
 214 :H275   = N275;       H276   = N276;       H277   = N277;
 215 :H301   = N301;       H302   = N302;       H303   = N303;
 216 :H304   = N304;       H305   = N305;       H306   = N306;
 217 :H307   = N307;       H308   = N308;       H309   = N309;
 218
 219 :
###
###  WARNING NO. 11
###  CDL2- 1 IC(s) with fast nail deficit.
###
###    93  H nails out of 3584  H nails have been allocated
###
```

7.3 RUN CIRCUIT EXTRACTOR

In addition to the .WOR file, for use by ATG and DTG, two other circuit data files are required: the In-circuit Diagnostic Data (.IDD) file and the Digital ATG Interconnect (.DAI) file. The .IDD file is used by DTG and the run-time system. The .DAI file is used by DTG. Enable "IDD prepare" on the ATG page to run the Circuit Description Extractor (CDE) task that creates the 2 files.

A report is generated that summarizes the data processed from the .CKT and used in the .DAI and .IDD files. This is the Extractor Report (.ERP) which should be carefully checked for concurrence with input data to avoid problems later.



```
Circuit Extractor Report for SC:TRAINGBD.WOR 27OCT83  10:33:45

    number of nails   = 93
    number of digital nodes      = 73
warning -- this ic not found in any library:  LM1458
warning -- this ic not found in any library:  RPAK8
warning -- this ic not found in any library:  NE5118
warning -- this ic not found in any library:  XTL
    number of undrivable ic's    = 0

IDD creation summary:
    number of device entries  = 51
    number of pin/nail entries  = 51
    number of nail entries      = 93
    number of nail/node entries = 93
```

The data summarized in this report is as follows:

Number of nails defined in the ADAPTOR section of the .CKT.

Number of nodes used in digital test.

Warnings for devices that have no approriate test in a system or update library.

Number of ICs that have inputs attached to a node that has an UNDRIV pin list in the HEADER section of a library routine.

Number of devices entered in the .IDD file. Should be the same as the number of devices in the CIRCUIT section of the .CKT.

Number of device pin to node lists. Should be the same as the number of devices because each device has a pin list in the CIRCUIT section of the .CKT.

Number of nails entered in the .IDD. Should equal number of nails in .CKT.

Number of nail to node entries in the .IDD. Each nail should be associated with a node.

Figure 7-4. Extractor Report (.ERP).

7.4 GENERATE CROSS REFERENCE REPORT (.AXR)

When ATG is run with "cross reference" selected on the option page, an ATG Cross Report (.AXR) is generated. "Circuit prepare" and "IDD prepare" options must have been run prior to this option. It is strongly recommended that the data in this report be thoroughly checked with the board input data to assure no circuit information is wrong in the test program. An error caught here will save much grief at debug time.

TRAINGBD.AXR

```
CIRCUIT CROSS REFERENCE AND DISCREPANCIES REPORT 26-OCT-83 15 11
For Board TRAINGBD

        *** DEVICE DATA *****

DEVICE  VALUE AND CONNECTIONS (R)

R1      1K
        1 = H309,  2 = H277
R12     5 1K
        1 = H118,  2 = H16
R13     10K
        1 = H77,  2 = H16
R14     3. 3K
        1 = H77,  2 = H54
R15     100K
        1 = H21,  2 = H79
R16     4 7K
        1 = H18,  2 = H81
R17     390K
        1 = H85,  2 = H84
R18     1MEG
        1 = H17,  2 = H85
R19     1         ,+  1. 0%, -  1 0%

,                                    RLORESID =  1
        1 = H46,  2 = H48
R2      1K
        1 = H277,  2 = H16
R20     3. 3K
        1 = H86,  2 = H49
R21     510K
        1 = H15,  2 = H50
R22     470K
        1 = H50,  2 = H51
R23     10K
        1 = H22,  2 = H115
R24     10MEG   ,+  5. 0%, -  5. 0%
        1 = H15,  2 = H117
R3      10K
        1 = H15,  2 = H115
R4      10K
        1 = H115,  2 = H113
R5      3. 3K
        1 = H15,  2 = H269
R6      3 3K
        1 = H15,  2 = H270
R7      3. 3K
        1 = H15,  2 = H307
R8      3. 3K
        1 = H15,  2 = H273
R9      330
        1 = H15,  2 = H272


DEVICE  VALUE AND CONNECTIONS (C)'

C2      10N
        1 = H53,  2 = H18
C3      20P
        1 = H181,  2 = H117
C4      1U
        1 = H84,  2 = H16
C5      10N
        1 = H83,  2 = H16
C6      10U
        1 = H51,  2 = H16
C7      10N
        1 = H52,  2 = H16


DEVICE  VALUE AND CONNECTIONS (RV)

R100 ,  10K     ,+ 20. 0%, - 20. 0%
        1 = H54,  2 = H54,  3 = H21


DEVICE  VALUE AND CONNECTIONS (CR)

CR1
        A = H309,  C = H16
CR2
        A = H79,  C = H81
CR4
        A = H16,  C = H82
DS1
        A = H80,  C = H46


DEVICE  VALUE AND CONNECTIONS (VZ)

CR3     7. 5
        A = H81,  C = H82
```

Comments

This report is generated when the "cross reference" option is run from the ATG monitor page. It lists all circuit data in a way that makes it easy to check for problems. These data are first presented under the ***DEVICE DATA*** heading, in groups, by device type. They are later presented under the heading *****NODE CONNECTIONS*****, by temporary nails, in the order that they appear in the .CKT.

The device data lists for each device, all data from the value section of the .CKT, then, on the next line, the temporary nail assigned to the node for each device lead. The complete TRAINGBD.AXR is listed here with descriptive comments beside it.

If a nail had been omitted in the .CKT for a node, an error message appears after the affected device lead. For example, connection R3.2 has no nail at the node. The listing for R3 would be

```
R13   10K
      1 = H77, 2 = N16
***** (external name) NO SCANNER PINS
      CONNECTED TO THIS PIN *****
```

where: N16 is the node name instead of a nail, such as H16.

All data provided in the VALUE section of the .CKT is written into the .AXR. So, any flagspecs appear as shown by the RLORESID = 1 at R19, the MSG flagspec at device CR3, and the RAIL flagspecs at U6.

If a value had been omitted in the .CKT for a device, an error message appears after that device. For example, R9 has no value. The listing for R9 would be

```
R9 ** MISSING VALUE **
1 = H15, 2 = H272
```

If a value had been given in the .CKT that was unlikely for the device, a warning message appears after that value. For example, C4 has the value 1000 mF. The listing for C4 would be

```
C4    1000 U
** UNUSUAL VALUE **
      1 = H84, 2 = H16
```

A list of all unusual values defined by the system is given in Table 7-3.

```
DEVICE  VALUE AND CONNECTIONS (QN) continued

Q1
        B = H277, C = H115, E = H16


DEVICE  VALUE AND CONNECTIONS (CP1)

C1      10U
        P = H15, N = H309


DEVICE  VALUE AND CONNECTIONS (LM1458)

U6                              NEGRAIL =  15,
                                POSRAIL =  15
        1 = H181, 2 = H117, 3 = H118, 4 = H18, 5 = H181, 6 = H77,
        7 = H21, 8 = H17


DEVICE  VALUE AND CONNECTIONS (RPAK8)

Z1
        1 = H15, 2 = H302, 3 = H304, 4 = H113, 5 = H143, 6 = H207,
        7 = H146, 8 = H211, 9 = H210


DEVICE  VALUE AND CONNECTIONS (7400)

U1
        1 = H301, 2 = H302, 3 = H303, 4 = H303, 5 = H304, 6 = H305,
        7 = H16, 8 = H110, 9 = H212, 10 = H111, 11 = H112, 12 = H110,
         13 = H111, 14 = H15


DEVICE  VALUE AND CONNECTIONS (7414)

U10
        1 = H115, 2 = H114, 3 = H114, 4 = H113, 5 = H209, 6 = H111,
        7 = H16, 8 = H13, 9 = H78, 10 = H19, 11 = H14, 12 = H116,
        13 = H20, 14 = H15

DEVICE  VALUE AND CONNECTIONS (2114)

U11
        1 = H176, 2 = H177, 3 = H178, 4 = H179, 5 = H142, 6 = H141,
        7 = H180, 8 = H110, 9 = H16, 10 = H109, 11 = H206, 12 = H205,
         13 = H244, 14 = H243, 15 = H173, 16 = H174, 17 = H175,
        18 = H15
U3
        1 = H176, 2 = H177, 3 = H178, 4 = H179, 5 = H142, 6 = H141,
        7 = H180, 8 = H110, 9 = H16, 10 = H109, 11 = H242, 12 = H241,
         13 = H240, 14 = H239, 15 = H173, 16 = H174, 17 = H175,
        18 = H15


DEVICE  VALUE AND CONNECTIONS (Z80CPU)

U2
        1 = H274, 2 = H275, 3 = H276, 4 = H237, 5 = H238, 6 = H272,
        7 = H243, 8 = H242, 9 = H244, 10 = H205, 11 = H15, 12 = H241,
         13 = H206, 14 = H239, 15 = H240, 16 = H210, 17 = H211,
        18 = H207, 19 = H209, 20 = H208, 21 = H144, 22 = H109,
        23 = H145, 24 = H143, 25 = H146, 26 = H113, 27 = H147,
        28 = H148, 29 = H16, 30 = H142, 31 = H141, 32 = H180,
        33 = H179, 34 = H178, 35 = H177, 36 = H176, 37 = H175,
        38 = H174, 39 = H173, 40 = H212


DEVICE  VALUE AND CONNECTIONS (2716)

U4
        1 = H175, 2 = H176, 3 = H177, 4 = H178, 5 = H179, 6 = H180,
        7 = H141, 8 = H142, 9 = H239, 10 = H240, 11 = H241, 12 = H16,
         13 = H242, 14 = H243, 15 = H244, 16 = H205, 17 = H206,
        18 = H112, 19 = H212, 20 = H112, 21 = H15, 22 = H173,
        23 = H174, 24 = H15


DEVICE  VALUE AND CONNECTIONS (NE5118)

U5
        1 = H16, 2 = H239, 3 = H240, 4 = H241, 5 = H242, 6 = H243,
        7 = H244, 8 = H205, 9 = H206, 10 = H208, 11 = H213, 12 = H245,
         13 = H16, 14 = H149, 15 = H245, 16 = H53, 17 = H18, 18 = H150,
         19 = H17, 20 = H181, 21 = H117, 22 = H16
```

TRAINGBD.AXR (continued)

Comments

```
DEVICE  VALUE AND CONNECTIONS (NE555) continued

U7
        1 = H16, 2 = H84, 3 = H82, 4 = H17, 5 = H83, 6 = H84,
        7 = H85, 8 = H18
U8
        1 = H16, 2 = H51, 3 = H49, 4 = H15, 5 = H52, 6 = H51,
        7 = H50, 8 = H15


DEVICE  VALUE AND CONNECTIONS (7474)

U9
        1 = H307, 2 = H306, 3 = H305, 4 = H269, 5 = H308, 6 = H306,
        7 = H16, 8 = H271, 9 = H272, 10 = H270, 11 = H308, 12 = H271,
         13 = H273, 14 = H15


DEVICE  VALUE AND CONNECTIONS (XTL)

Y1
        7 = H16, 8 = H301, 14 = H15


DEVICE  VALUE AND CONNECTIONS (SCR)

Q3
        C = H16, A = H48, G = H86


DEVICE  VALUE AND CONNECTIONS (NJFET)

Q2
        S = H21, G = H79, D = H80


DEVICE  VALUE AND CONNECTIONS (NJFET)

Q2
        S = H21, G = H79, D = H80

NO MISSING VALUES
```

If any device in the .CKT did not have a required value specified, a message would be included here in place of this message.

```
NO UNUSUAL VALUES
```

If any device in the .CKT had a value specified that was unusual for the device, a message would be included here in place of this message.

```
                    ***** NODE CONNECTIONS *****

SCANNER PIN  NODE NAME  OTHER SCANNER PINS AND DEVICE PINS

GND      =   N16        H16, C4.2, C5.2, C6 2, C7.2, CR1 C, CR4 A,
                        Q1.E, R12.2, R13.2, R2.2, Q3.C, U1.7, U10 7,
                         U11.9, U2 29, U3 9, U4.12, U5.1, U5.13,
                        U5 22, U7.1, U8.1, U9 7, Y1.7

VCC      =   N15        H15, C1.P, R21 1, R24.1, R3.1, R5 1, R6.1,
                        R7.1, R8.1, R9.1, Z1.1, U1.14, U10.14,
                        U11.18, U2.11, U3 18, U4 21, U4 24, U8 4,
                        U8.8, U9.14, Y1.14

VCC15    =   N17        H17, R18.1, U6.8, U5 19, U7.4
VCC15N   =   N18        H18, C2.2, R16.1, U6.4, U5 17, U7.8
H13      =   N13        U10.8
H14      =   N14        U10.11
H15      =   N15        VCC, C1.P, R21.1, R24.1, R3.1, R5.1, R6.1,
                        R7 1, R8.1, R9 1, Z1.1, U1.14, U10.14,
                        U11.18, U2.11, U3.18, U4 21, U4.24, U8.4,
                        U8 8, U9.14, Y1.14

H16      =   N16        GND, C4.2, C5.2, C6.2, C7.2, CR1.C, CR4 A,
                        Q1.E, R12.2, R13.2, R2 2, Q3.C, U1.7, U10.7,
                         U11.9, U2 29, U3 9, U4.12, U5.1, U5.13,
                        U5.22, U7.1, U8.1, U9.7, Y1.7
```

The Node Connection section lists every external node with all the adaptor nails and circuit device pins that are connected to it. Then every adaptor nail is listed with all the external node names and device pins that are associated with it. Any nodes with no scanner pins will be listed at the end.

TRAINGBD.AXR (continued)

Comments

```
SCANNER PIN  NODE NAME  OTHER SCANNER PINS AND DEVICE PINS

 H17      =     N17        VCC15, R18.1, U6.8, U5.19, U7.4
 H18      =     N18        VCC15N, C2.2, R16.1, U6.4, U5.17, U7.8
 H19      =     N19        U10.10
 H20      =     N20        U10.13
 H21      =     N21        R15.1, R100.3, Q2.S, U6.7
 H22      =     N22        R23.1
 H46      =     N45        DS1.C, R19.1
 H48      =     N47        R19.2, Q3.A
 H49      =     N49        R20.2, U8.3
 H50      =     N50        R21.2, R22.1, U8.7
 H51      =     N51        C6.1, R22.2, U8.2, U8.6
 H52      =     N52        C7.1, U8.5
 H53      =     N53        C2.1, U5.16
 H54      =     N54        R14.2, R100.1, R100.2
 H77      =     N77        R13.1, R14.1, U6.6
 H78      =     N78        U10.9
 H79      =     N79        CR2.A, R15.2, Q2.G
 H80      =     N80        DS1.A, Q2.D
 H81      =     N81        CR2.C, R16.2, CR3.A
 H82      =     N82        CR4.C, CR3.C, U7.3
 H83      =     N83        C5.1, U7.5
 H84      =     N84        C4.1, R17.2, U7.2, U7.6
 H85      =     N85        R17.1, R18.2, U7.7
 H86      =     N86        R20.1, Q3.G
 H109     =     N109       U11.10, U2.22, U3.10
 H110     =     N110       U1.8, U1.12, U11.8, U3.8
 H111     =     N111       U1.10, U1.13, U10.6
 H112     =     N112       U1.11, U4.18, U4.20
 H113     =     N113       R4.2, Z1.4, U10.4, U2.26
 H114     =     N114       U10.2, U10.3
 H115     =     N115       Q1.C, R23.2, R3.2, R4.1, U10.1
 H116     =     N116       U10.12
 H117     =     N117       C3.2, R24.2, U6.2, U5.21
 H118     =     N118       R12.1, U6.3
 H141     =     N141       U11.6, U2.31, U3.6, U4.7
 H142     =     N142       U11.5, U2.30, U3.5, U4.8
 H143     =     N143       Z1.5, U2.24
 H144     =     N144       U2.21
 H145     =     N145       U2.23
 H146     =     N146       Z1.7, U2.25
 H147     =     N147       U2.27
 H148     =     N148       U2.28
 H149     =     N149       U5.14
 H150     =     N150       U5.18
 H173     =     N173       U11.15, U2.39, U3.15, U4.22
 H174     =     N174       U11.16, U2.38, U3.16, U4.23
 H175     =     N175       U11.17, U2.37, U3.17, U4.1
 H176     =     N176       U11.1, U2.36, U3.1, U4.2
 H177     =     N177       U11.2, U2.35, U3.2, U4.3
 H178     =     N178       U11.3, U2.34, U3.3, U4.4
 H179     =     N179       U11.4, U2.33, U3.4, U4.5
 H180     =     N180       U11.7, U2.32, U3.7, U4.6
 H181     =     N181       C3.1, U6.1, U6.5, U5.20
 H205     =     N205       U11.12, U2.10, U4.16, U5.8
 H206     =     N206       U11.11, U2.13, U4.17, U5.9
 H207     =     N207       Z1.6, U2.18
 H208     =     N208       U2.20, U5.10
 H209     =     N209       U10.5, U2.19
 H210     =     N210       Z1.9, U2.16
 H211     =     N211       Z1.8, U2.17
 H212     =     N212       U1.9, U2.40, U4.19
 H213     =     N213       U5.11
 H237     =     N237       U2.4
 H238     =     N238       U2.5
 H239     =     N239       U2.14, U3.14, U4.9, U5.2
 H240     =     N240       U2.15, U3.13, U4.10, U5.3
 H241     =     N241       U2.12, U3.12, U4.11, U5.4
 H242     =     N242       U2.8, U3.11, U4.13, U5.5
 H243     =     N243       U11.14, U2.7, U4.14, U5.6
 H244     =     N244       U11.13, U2.9, U4.15, U5.7
 H245     =     N245       U5.12, U5 15
 H269     =     N269       R5.2, U9.4
 H270     =     N270       R6.2, U9.10
 H271     =     N271       U9.8, U9.12
 H272     =     N272       R9.2, U2.6, U9.9
 H273     =     N273       R8.2, U9.13
 H274     =     N274       U2.1
 H275     =     N275       U2.2
 H276     =     N276       U2.3
 H277     =     N277       Q1.B, R1.2, R2.1
 H301     =     N301       U1.1, Y1.8
 H302     =     N302       Z1.2, U1.2
 H303     =     N303       U1.3, U1.4
 H304     =     N304       Z1.3, U1.5
 H305     =     N305       U1.6, U9.3
 H306     =     N306       U9.2, U9.6
 H307     =     N307       R7.2, U9.1
 H308     =     N308       U9.5, U9.11
 H309     =     N309       C1.N, CR1.A, R1.1


 NO NODES WITHOUT SCANNER PINS
```

Table 7-3
LIST OF UNUSUAL VALUES

Device	Value less than	 greater than
Resistor (R)	1.0 ohm	10 megohms
Capacitor (C)	10 pF	10 uF
Polarized Capacitor (CP)	10 nF	10,000 uF
Polarized Capacitor (CP1)	10 nF	1,000 uF
Inductor (L)	10 uH	1.0 H
Potentiometer (RV)	10 ohms	10 megohms
Jumper (J)	0 ohm	1.0 ohm
Fuse (F)	0.1 ohm	10 ohms
Diode (CR)	0.1 volt	1.2 volts
Zener Diode (VZ)	1.0 volt	200 volts
NPN Transistor (QN)	gain = 5	gain = 100,000
PNP Transistor (QP)	gain = 5	gain = 100,000
IC	0.1 volt	1.2 volts

7.5 RUN ANALOG TEST GENERATION

Once there is an error-free compiled circuit description file (.WOR), the analog phase of ATG can begin. The ATG monitor page will appear on the video display after the compile has been successfully completed. The appropriate "Analog test gen." option can be selected. If a User Analog Library file is to be used, the LIBuser option must be set to [Y] and the filespec must be stated in the "LIB =" option field. During this phase, ATG will develop shorts tests and opens tests between nodes in the circuit and will attempt to generate a test for each analog, complex or hybrid component in the circuit description.

Any analog, complex or hybrid components that could not be tested by an analog library test procedure will be noted in the analog report file. If ATG has enough information, a FORM Test is also included in the test program and is preceded by a branch statement that causes the FORM test to be bypassed. When this is not possible, ATG will write a much simpler dummy test near the beginning of the test program.

An analysis of a circuit with untested components should result in decisions about modifying the test program. Some of the decisions may be:

Use the FORM test inserted by ATG.
Change flagspecs in the .CKT file.
Modify the analog library procedures.
Insert user-designed test.
Do not try to test the component.

If the FORM test is to be used, the .ATP must be edited to provide any new data needed for the FORM test and to remove the branch statement that precedes the FORM test that would otherwise cause the test to be bypassed. An example of the branch statement is R7: BRANCH R7Z;, where R7 is the component name for which a FORM test was inserted and R7Z is the label immediately after the test. FORM tests usually have comments in them which instruct the user about the required edits. These edits are most easily done using debug commands described in Section 9.

Refer to Section 6 of this manual for flagspec information if the decision is made to change flagspecs. Refer to the 227X Test Library Programming manual if the decision is made to change the library. Refer to the 227X Test Programming manual if a decision to write a new test is made.

The ATG task generates an analog ATG report (.RPT) which provides data on each component for which ATG tried to generate a test. The report also gives statistics on the quality of tests found by component types. The report is organized so that all tests and devices of one type are grouped together.

The complete TRAINGBD.RPT and TRAINGBD.ATP follow with descriptive comments inserted beside them.

TRAINGBD.RPT

```
ANALOG ATG(ATGMN419) REPORT FOR TRAINGBD          31-OCT-83 11:48
        PAGE 1


Libraries used during analog ATG
--------------------------------
Library SYS: LB1:[2,11]TSTLIB740.LIB


Connectivity results
-------------------------------------------------------------------------
Lib  Test     Page

SYS  OPENS    4.1

Pin list: 45,46: 0
Analog threshold:         10 ohms
Digital threshold:        45 ohms
Test voltage:             100M

SYS  OPENS    4.1

Pin list: 47,48: 0
Analog threshold:         10 ohms
Digital threshold:        45 ohms
Test voltage:             100M

SYS  NOPSHRT  19.1

Pin list: 15-18: 0: 0
Analog threshold:         10 ohms
Digital threshold:        45 ohms
Test voltage:             100M

SYS  NOSHRT   20.1

Pin list: 13-22,45,49-54,77-86,109-118,141-150,173-181,205-213,239-242,
        244,245,269-273,277,301-309,4001,4002,4009,4010,4017,4018: 0
         0
Analog threshold:         10 ohms
Digital threshold:        45 ohms
Test voltage:             100M
```

Comments

The report starts by identifying the board and the libraries used for analog ATG.

The following connectivity tests were generated, and reported on this page:

1. An OPENS test was written from the SYStem library to stream 4 page 1 of the .TPX that checks for opens between the two temp nails (45 and 46) that were assigned to node N45. The threshold resistance used as a result of the default or a flagspec and the test voltage used, are shown in the report.
2. Test for an open between the two temp nails (47 and 48) that were assigned to node N47.
3. Test for a power short between all power and ground nails as follows:

VCC (nail 15)	GND (nail 16)
VCC15 (nail 17)	VCCN15 (nail 18)

4. Test for a short between any of the remaining nails.

```
ANALOG ATG(ATGMN419) REPORT FOR TRAINGBD          31-OCT-83 11.48
        PAGE 2
C Component results
-------------------------------------------------------------------------
Name    Value      Tols%  Err%   Lib Directory Test    Result   Page

C2      10N            5  1.18   SYS CAP       CGRD      S      42.1


C3      20P            5  45.9   SYS CAP       CGRD      A      42.1
     LIMITS CONTAIN SYSTEM RESIDUALS
     LARGE INSTRUMENT MEASUREMENT ERROR
     LARGE ERROR DUE TO RESIDUAL VARIATIONS
     CONSIDER SPECIFYING "CHIRESID" & "CLORESID"
C4      1U             5  1.28   SYS CAP       CGRD      S      42.1


C5      10N            5  1.27   SYS CAP       CGRD      S      42.1


C6      10U            5  1.28   SYS CAP       CGRD      S      42.1


C7      10N            5  1.27   SYS CAP       CGRD      S      42.1
```

This page of the .RPT lists information about tests written for testing capacitors. For example: the name of the first component is C2, its value is 10 nano farads, the tolerance rating is 5%, the measurement error calculated by ATG is 1.18%, the test procedure was taken from the SYStem library, the library directory and test name is CAP and CGRD, the quality of the test is superior and the test was written to stream 42, page 1 of the .ATP.

Capacitor C3 is a very low value (20 pico farad) and the test produced has a measurement error or 45.9% because of relation to equipment residual capacitances and instrument accuracy. The messages included with C3 recommend using the flagspecs CHIRESID and CLORESID to improve accuracy.

Component Test Statistics are included for each component type. The test statistics for TRAINGBD capacitors shows that ATG wrote 5 superior capacitance tests. The average error for these 5 tests was 1.26% and the 5 tests represent 83.33% of all "C" type capacitors processed by ATG. One adequate test was written and there were no capacitors that ATG could not find a test for, or that were untestable.

TRAINGBD.RPT (continued)

Comments

```
ANALOG ATG(ATGMN419) REPORT FOR TRAINGBD          31-OCT-83 11:48
       PAGE 3
CP1 Component results
-------------------------------------------------------------------------
Name      Value      Tols%  Err%   Lib Directory Test    Result    Page

C1        10U           20  1.76   SYS TANT      CPGRD     S       44.1



Component Test Statistics
      1 Superior Tests   (Average err:   1.76%)    (  100%)
      0 Adequate Tests
      0 Unsuccessful Tests
      0 Not Testable Characteristics
      1 Total Component Tests Processed
```

C1 is a polarized capacitor, which uses a type CP test. A guarded test was selected, which ATG puts in stream 44.

```
ANALOG ATG(ATGMN419) REPORT FOR TRAINGBD          31-OCT-83 11:48
       PAGE 4
CR Component results
-------------------------------------------------------------------------
Name      Value      Tols%  Err%   Lib Directory Test    Result    Page

CR1       650M                     SYS CRFORWD   CRFDFOR  U        52.1
      SHUNT CURRENT(SOURCE ON CATHODE) I=1.184MA
      SHUNT CURRENT(SOURCE ON ANODE) I=955UA
     CANNOT PUT SOURCE ON CATHODE.
     POSSIBLE DIODE FROM CATHODE TO GUARD.
     CANNOT PUT SOURCE ON ANODE.
     POSSIBLE DIODE FROM ANODE TO GUARD.
     Device CR1 Directory CRFORWD test not done
CR2       650M                     SYS CRFORWD   CRFWDG   S        51.1

CR4       650M                     SYS CRFORWD   CRFDFOR  U        52
      SHUNT CURRENT(SOURCE ON CATHODE) I=9.645UA
      SHUNT CURRENT(SOURCE ON ANODE) I=1.184MA
     ALL EXISTING TEST CONFIGURATIONS WERE VETOED
     CANNOT PUT SOURCE ON CATHODE.
     POSSIBLE DIODE FROM CATHODE TO GUARD.
     CANNOT PUT SOURCE ON ANODE.
     POSSIBLE DIODE FROM ANODE TO GUARD.
     Device CR4 Directory CRFORWD test not done
DS1       650M                     SYS CRFORWD   CRFWDG   S        51.1



Component Test Statistics
      2 Superior Tests                             (   50%)
      0 Adequate Tests
      2 Unsuccessful Tests                         (   50%)
      0 Not Testable Characteristics
      4 Total Component Tests Processed
```

ATG is unable to find a safe test for CR1 because of transistor Q1 being connected between the guard for CR1 and the cathode of CR1. Since ATG considers this connection a shunting diode, it concludes that it is unsafe to put the source on either the cathode or anode. Thus, ATG is unsuccessful in writing a test. The calculated shunt currents are given to help the user analyze the situation.

ATG is unable to find a safe test for CR4 for similar reasons that it could not do so for CR1. In this case U7 is the shunting device because ATG assumes, for circuit analysis, that a back-to-back diode model exists between all terminals of an IC.

```
ANALOG ATG(ATGMN419) REPORT FOR TRAINGBD          31-OCT-83 11:48
       PAGE 5
QN Component results
-------------------------------------------------------------------------
Name      Value      Tols%  Err%   Lib Directory Test    Result    Page

Q1        5                        SYS QGAIN     QB        S       71.1



Component Test Statistics
      1 Superior Tests                             (  100%)
      0 Adequate Tests
      0 Unsuccessful Tests
      0 Not Testable Characteristics
      1 Total Component Tests Processed
```

TRAINGBD.RPT (continued)

```
ANALOG ATG(ATGMN419) REPORT FOR TRAINGBD          31-OCT-83 11:48
          PAGE 6
R Component results
----------------------------------------------------------------------------
Name      Value       Tols%  Err%    Lib Directory Test    Result    Page

R1        1K              5  4.87    SYS RES       RGRD      S       26.1

R12       5.1K            5  1.21    SYS RES       RGRD      S       26.1

R13       10K             5  1.33    SYS RES       RGRD      S       26.1

R14       3.3K            5  1.11    SYS RES       RGRD      S       26.1

R15       100K            5  1.29    SYS RES       RGRD      S       26.1

R16       4.7K            5  1.05    SYS RES       RGRD      S       26.1

R17       390K            5  1.97    SYS RES       RGRD      S       26.1

R18       1MEG            5  3.61    SYS RES       RGRD      S       26.1

R19

      Component R19 from ATO
R2        1K              5  1.29    SYS RES       RGRD      S       26.1

R20       3.3K            5  1.03    SYS RES       RGRD      S       26.1

R21       510K            5  2.34    SYS RES       RGRD      S       26.1

R22       470K            5   2.2    SYS RES       RGRD      S       26.1
```

```
ANALOG ATG(ATGMN419) REPORT FOR TRAINGBD          31-OCT-83 11:48
          PAGE 7
R Component results
----------------------------------------------------------------------------
Name      Value       Tols%  Err%    Lib Directory Test    Result    Page
R23       10K             5   0.5    SYS RES       2TR       S       27.1

R24       10MEG           5    33    SYS RES       RGRD      A       26.2
      TOTAL CALCULATED ERRORS EXCEED GOOD TEST LIMIT
      I MEASURE OFFSET LIMITING ACCURACY
      LOW RESOLUTION DUE TO LOW VOLTAGE OR LARGE RESISTANCE
      INSTRUMENT ACCURACY LIMITING TEST

R3        10K             5  1.05    SYS RES       RGRD      S       26.2

R4        10K             5  1.03    SYS RES       RGRD      S       26.2

R5        3.3K            5  1.08    SYS RES       RGRD      S       26.2

R6        3.3K            5  1.08    SYS RES       RGRD      S       26.2

R7        3.3K            5  1.08    SYS RES       RGRD      S       26.2

R8        3.3K            5  1.08    SYS RES       RGRD      S       26.2

R9        330             5  1.23    SYS RES       RGRD      S       26.2


Component Test Statistics
     20 Superior Tests  (Average err:  1.57%)   (95.24%)
      1 Adequate Tests  (Average err: 33.01%)   (4.762%)
      0 Unsuccessful Tests
      0 Not Testable Characteristics
     21 Total Component Tests Processed
```

Comments

Most resistors are tested with a guarded resistance test and are put in stream 26.

The test for R19 is provided by the ATO file and no statistics are generated by ATG.

R23 is a resistor connected to an external point and therefore is tested with a 2 terminal test.

R24 is a very high value resistor which could only be tested with an accuracy of 33%. Therefore, its test is classed as adequate.

```
ANALOG ATG(ATGMN419) REPORT FOR TRAINGBD         31-OCT-83 11:48
        PAGE 8
RV Component results
-----------------------------------------------------------------------
Name     Value     Tols%  Err%   Lib Directory Test    Result   Page

R100     10K          20  0.38   SYS POT       RVADJG   S       16.1
     THIS POT TESTED AS ADJUSTED VARIABLE RESISTOR


Component Test Statistics
      1 Superior Tests  (Average err:  0.38%)  (  100%)
      0 Adequate Tests
      0 Unsuccessful Tests
      0 Not Testable Characteristics
      1 Total Component Tests Processed
```

```
ANALOG ATG(ATGMN419) REPORT FOR TRAINGBD         31-OCT-83 11:48
        PAGE 9
VZ Component results
-----------------------------------------------------------------------
Name     Value     Tols%  Err%   Lib Directory Test    Result   Page

CR3      7.5         10          SYS ZENER     VZFORM   U       87.1

     Device CR3 Directory ZENER test not done


Component Test Statistics
      0 Superior Tests
      0 Adequate Tests
      1 Unsuccessful Tests                         (  100%)
      0 Not Testable Characteristics
      1 Total Component Tests Processed
```

```
ANALOG ATG(ATGMN419) REPORT FOR TRAINGBD         31-OCT-83 11:48
        PAGE 10
SCR Component results
-----------------------------------------------------------------------
Name     Value     Tols%  Err%   Lib Directory Test    Result   Page

Q3                               SYS THYRIST   SCRIM    S       81.1


  CR1

     Device CR1 Test not considered (NOTEST attribute found)


Component Test Statistics
      1 Superior Tests                             (  100%)
      0 Adequate Tests
      0 Unsuccessful Tests
      0 Not Testable Characteristics
      1 Total Component Tests Processed
```

```
ANALOG ATG(ATGMN419) REPORT FOR TRAINGBD         31-OCT-83 11:48
        PAGE 11
NJFET Component results
-----------------------------------------------------------------------
Name     Value     Tols%  Err%   Lib Directory Test    Result   Page

Q2       2M                      SYS TRANCON   JFETFOR  S       79.1


  SG

     Device SG Test not considered (NOTEST attribute found)
  DG

     Device DG Test not considered (NOTEST attribute found)
  SD

     Device SD Test not considered (NOTEST attribute found)


Component Test Statistics
      1 Superior Tests                             (  100%)
      0 Adequate Tests
      0 Unsuccessful Tests
      0 Not Testable Characteristics
      1 Total Component Tests Processed
```

TRAINGBD.RPT (continued)

```
ANALOG ATG(ATGMN419) REPORT FOR TRAINGBD        31-OCT-83 11:48
        PAGE 12
IC Component results
-------------------------------------------------------------------------
Name     Value     Tols%  Err%   Lib Directory Test   Result    Page

U6

     Component U6 from ATD


Component Test Statistics
      0 Superior Tests
      0 Adequate Tests
      0 Unsuccessful Tests
      0 Not Testable Characteristics
      0 Total Component Tests Processed




ANALOG ATG(ATGMN419) REPORT FOR TRAINGBD        31-OCT-83 11:48
        PAGE 13
Utility results
-------------------------------------------------------------------------
Lib  Test

SYS  TPDEC

SYS  TPMUX

SYS  TPEND

SYS  TPSUB


Board Test Statistics
     32 Superior Tests  (Average err:  1.25%)   (86.49%)
      2 Adequate Tests  (Average err: 39.43%)   (5.405%)
      3 Unsuccessful Tests                      (8.108%)
      0 Not Testable Characteristics
     10 Connectivity, Utility, and ATD Tests
     47 Total Board Tests Processed
```

Comments

Statistics for the group of components are listed below the component data. The statistics include:

1. Total number of components processed by ATG.
2. Number of component tests that resulted in each of the result categories (S, A, U, N).
3. The average test error calculated by ATG for all tested components.
4. The percentage of components in each test result category.

If a NOTEST flagspec was selected, the following message will appear:

Test not considered (NOTEST attribute found)

If no test in any test library is found for advice, the message will be:

Test not considered (None in libraries)

For complex components, the NAME column contains the internal names from the component library next to their corresponding tests. These data will not appear for devices in the .CKT file unless they belong to the primitive analog types (excluding the type IC) or the types in the analog component library.

TRAINGBD.ATP

Comments

```
        /* Target system: 2275  Adaptor: 2275  */
/* ATGMN419 TRAINGBD        31-OCT-83 11:48         PAGE 1.1 */

        DECLARE PWRDN();
                         DECLARE FAILCRLED ();
                         DECLARE LOOP;
DECLARE GROUP DACDATA(206,205,244,243,242,241,240,239);
DECLARE FLOAT DACOUT1;          DECLARE FLOAT DACOUT3;
DECLARE FLOAT DACOUT2;          DECLARE FLOAT DACOUT4;
DECLARE FLOAT VREFOUT;




/* THIS PROGRAM WAS GENERATED USING CIRCUIT DESCRIPTION TRAINGBD */

 DECLARE RVAL;                  DECLARE CBETA;
 DECLARE VVAL1;                 DECLARE VVAL2;
 DECLARE IVAL1;                 DECLARE IVAL2;
 DECLARE CSTRING DSTRG(20);     DECLARE BSTRING FAIL(5);
 DECLARE VGS1;  /* JFET GATE TO SOURCE VOLTAGE ONE */
 DECLARE VGS2;  /* JFET GATE TO SOURCE VOLTAGE TWO */
 DECLARE GM;    /* CALCULATED JFET TRANSCONDUCTANCE */


 DECLARE FAILCR();
 DECLARE LOAD FLOAT PILIM;      DECLARE LOAD FLOAT VSFAIL;
 DECLARE LOAD FLOAT NILIM;      DECLARE LOAD GETILIM();
 DECLARE FAILSUB();
 DECLARE FAILQDCS();            DECLARE FAILXTOR();

 DECLARE VRADJ(FLOAT RVAL, FLOAT HIGH, FLOAT LOW, FLOAT I,
        FLOAT DLY, FLOAT RDLY, FLOAT VOLT, CSTRING NAME(8),
        CSTRING MSG(72));

 DECLARE FAILCOMP();

ENDDEC:
```

The analog test program starts by identifying the target system and fixture adaptor for which ATG is generating the program. Then the page header is written which identifies the ATG task (AGMN419), the board name, the generation date and the page. The page number consists of the stream number and the page number within the stream. Stream 1 contains all the DECLARE statements. Several statements are added to the declare stream by the ATO file. Since the ATO file inserted these statements with an INCLUDE BEFORE command, the added statements are at the beginning of the stream ahead of the DECLARE statements that ATG automatically inserts.

```
/*       CHECK TO SEE IF THE DC SOURCE FAULTY CURRENT LIMIT FLAG
        WAS SET IN THE "GETILIM" LOAD SUBROUTINE.                */

IF VSFAIL = 1 THEN BRANCH PDONE;
```

At run-time, when the test program is loaded into computer memory, any LOAD routines that are in the program are automatically done. This program contains a GETILIM LOAD routine which checks proper DC Source operation. The first instruction of the program checks the result and branches to the end if the check failed.

```
/* ATGMN419 TRAINGBD        31-OCT-83 11:48         PAGE 4.1 */

        /* OPEN'S TEST AT MULTI-NAILED NODES */
T21:    TEST OPENS AT(45,46:0);
T41:    TEST OPENS AT(47,48:0);
```

The next test is done wherever there are more than one nail on a node. It checks for continuity between all nails on a node. The test for R19 requires 2 nails on each node.

```
/* ATGMN419 TRAINGBD        31-OCT-83 11:48         PAGE 14.1 */

SETRV:  /* THIS IS THE SET UP FOR ADJUSTING VARIABLE RESISTORS.
        UNLESS OTHERWISE SPECIFIED, DEFAULTS PROVIDE..

                V=200MV   I=128MA      DLY=0   RDLY=0   FAIL(1)

        CHA --- VOLTAGE SOURCE
        CHB --- CURRENT MEASURE
        CHC --- GUARD .SENSE
        CHD --- GUARD

        SET SCAN AT(SOURCE: MEASURE: GUARD SENSE: GUARD)

                                                                *⁄


        SET MUX AT(CHA=6: CHB=8: CHC=9: CHD=10: REF=5,6,11);
        USE FSUB=FAILSUB,
```

When there are potentiometers in a circuit, it is important to adjust them so that they are not in the shorted position. Therefore, the potentiometer tests are performed before the shorts testing. R100 is connected in the circuit as a variable resistor. The set-up for variable resistors is put in stream 14. The actual test for variable resistors is put in stream 16.

TRAINGBD.ATP (continued)

```
/* ATGMN419 TRAINGBD     31-OCT-83 11:48         PAGE 16.1 */

R100_T: SET SCAN AT (CHA=54:CHB=21:CHC=77:CHD=77);
        MEAS R INTO RVAL V=1 I=128M MAX=10K
        DLY=0 RDLY=0 FAIL();
        IF RVAL > 7.971K THEN IF RVAL < 12.041K THEN
                BRANCH R100E;
        CALL VRADJ(RVAL=10K,HIGH=12.041K,LOW=7.971K,
        I=128M,DLY=0,RDLY=0,VOLT=1,
        NAME='R100',MSG='');
R100:   MEAS R V=1 I=128M HI=12.041K LO=7.971K MAX=10K
        DLY=0 RDLY=0 DMSG='10K' [
                WRITE 'R100 (10K OHMS) FAILED VALUE=%G%%033%K
'                       FMEAS;
                BRANCH R100E;
                ];
        WRITE 'POT R100 ADJUSTED %033%K %NL%';
R100E:
```

```
/* ATGMN419 TRAINGBD     31-OCT-83 11:48         PAGE 19.1 */

        /* POWER/GROUND SHORT TEST */
        USE NOFSUB;
PWRGND: TEST SHORTS AT(15-18:0:0)
         VMAX=100M R=10  FAIL(4);
```

```
/* ATGMN419 TRAINGBD     31-OCT-83 11:48         PAGE 20.1 */

        /* SHORTS TEST */
        USE NOFSUB;
PRINT:  TEST SHORTS AT( 13-22,45,49-54,77-86,109-118,141-150,173-181,205-213,
        239-242,244,245,269-273,277,301-309,4001,4002,4009,4010,4017,4018:
         0: 0)
         VMAX=100M R=10
         FAIL(3) [WRITE 'FAILED SHORTS TESTS - TESTING HALTED%NL%';
                BRANCH DONE; ];
        /* POWER/GROUND SHORT */
        IF FAIL(4) THEN [WRITE 'POWER & GROUND SHORTED - ';
                        WRITE 'TESTING HALTED%NL%';
                        BRANCH FDONE; ];
        /* CONTINUE ON SHORTS */
        IF FAIL(3) THEN [WRITE 'FAILED SHORTS TESTS - ';
                        WRITE 'CONTINUE WITH DIGITAL%NL%';
                        BRANCH ADONE; ];
```

```
REPLACE R19 (39)    [

/* =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
        */
/* KELVIN MEASUREMENT PROCEEDURE FOR R19 */
/* =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

        */

/* IN THIS PROCEEDURE, THE MULTIPLEXER IS CONFIGURED WITH THE DC VOLTAGE
 MEASUREMENT UNIT ACROSS THE DEVICE UNDER TEST WITH ITS REFERENCE LINE
 LEFT FLOATING. THIS IS DONE BY PUTTING (P+) ON CHC AND (P-) ON CHD.
 THEN THE DC VOLTAGE SOURCE (VHI) IS CONNECTED TO CHA. THE DC CURRENT
 MEASUREMENT UNIT (IIN) IS THEN CONNECTED TO CHB. THE REFERENCE LINES
 OF THE DCS (REF5 AND REF6) ARE THEN CONNECTED TO THE REFERENCE LINE OF
 THE DC CURRENT MEASUREMENT UNIT (REF8) AND CONNECTED TO INSTRUMENTATION
 GROUND WITH REF-11. THIS TEST PROCEEDURE REQUIRES TWO TESTER NAILS ON
 EACH SIDE OF THE DUT. ONE PAIR OF NAILS ARE CONNECTED TO THE DC VOLTAGE
 MEASUREMENT UNIT AND THE OTHER PAIR OF NAILS ARE USED TO CARRY THE TEST
 CURRENT  THE CURRENT FLOW THROUGH THE INSTRUMENTATION AND THE DUT IS
 MEASURED AS IS THE VOLTAGE DROP ACCROSS THE DUT  USING THE FORMULA


 RDUT = VDUT / I
 THE ACTUAL VALUE OF THE DUT IS MEASURED ACCURATLY BY BYPASSING THE
 NORMALLY ENCOUNTERED SYSTEM RESIDUALS  */

        /*KELVIN MEAS FOR 1 OHM */
        USE FSUB=FAILSUB;
        R19     SET MUX AT(CHA=6 CHB=8 CHC=4 CHD=5 REF=5,6,8,11)
                SET SCAN AT(48 46 47 45);
                SET DCV DCS V=400M I=128M DLY=10M
                MEAS DCV INTO VVAL1 DLY=30M,
                MEAS DCI INTO IVAL1 DLY=30M,
                LET VVAL2=(VVAL1/IVAL1),
                TEST ARITH VVAL2 HI=1.33 LO 0.9 CMSG='R19' DMSG='1 OHM

         ,
        /* END OF KELVIN TEST */
]
```

Comments

Next the test for shorts between any two nails (except nails on the same node) is done. The set-up is in stream 19 and the test in stream 20.

Tests for resistors is done next in streams 24, 26 and 27. these are omitted.

The Kelvin test for R19, that was inserted from the ATO file, is next. Stream 39 was created for this test.

Tests in streams from 40 through 85 are omitted.

```
/* ATGMN419 TRAINGBD     31-OCT-83 11:48        PAGE 87.1 */

CR3_C:  BRANCH CR3_CA;  /* CATHODE ON SOURCE, UNGUARDED TEST */
        SET SCAN AT(82:82:81:81);
        SET DCV DCS V=10.449 I=20M;
/* IF V=16, THE REQUIRED VOLTAGE IS PROBABLY OVER 16V (INSTRUMENT MAX).  */

/*      SET DCV DCS V=10.449 I=128M;*/
/*  THE ABOVE STATEMENT USES 128 MA TEST CURRENT WHICH MAY BE NECESSARY
TO PROVIDE ADEQUATE CURRENT IF SHUNT RESISTANCE IS LOW.
---CAUTION---MAY DAMAGE SHUNTING DEVICES OR DUT, ESPECIALLY IF SHUNTING
RESISTIVE COMPONENTS ARE MISSING.  */

        MEAS DCV DCM
         CMSG='CR3' DMSG='7.5' HI=8.25 LO=6.75;
        SET DCV DCS V=0;
        DELAY 0;        /* MAY BE NECESSARY TO DISCHARGE CAPACITORS.  */
CR3_CA.
CR3_A:  BRANCH CR3_AN;  /* SOURCE ON ANODE, UNGUARDED TEST */
        SET SCAN AT(81:81:82:82);
        SET DCV DCS V=-10.449 I=20M;
/* IF V=16, THE REQUIRED VOLTAGE IS PROBABLY OVER 16V (INSTRUMENT MAX)   */

/*      SET DCV DCS V=10.449 I=128M;*/
/*  THE ABOVE STATEMENT USES 128 MA TEST CURRENT WHICH MAY BE NECESSARY
TO PROVIDE ADEQUATE CURRENT IF SHUNT RESISTANCE IS LOW
---CAUTION---MAY DAMAGE SHUNTING DEVICES OR DUT, ESPECIALLY IF SHUNTING
RESISTIVE COMPONENTS ARE MISSING.  */


        MEAS DCV DCM
         CMSG='CR3' DMSG='7.5' HI=-8.25 LO=-6.75;
        SET DCV DCS V=0;
        DELAY 0;        /* MAY BE NECESSARY TO DISCHARGE CAPACITORS.  */
CR3_AN:

        WRITE 'NOW TURNING UP POWER%NL%',

        ON ABORT CALL PWRDN();
PWRUP:  SET PIO(0) HRLY(CLOSE 3);       /* GROUND PCB */
        SET PIO(0) HRLY(CLOSE 6); /* CONNECT FIXED P.S.  TO DUT */
        DELAY 10M;
        SET PIO(0) RLY(CLOSE 12);       /* TURN ON FIXED P.S.  OUTPUTS */
        DELAY 10M;
        SET MUX AT (4:5);               /* CONNECT DCVM TO MUX */

        SET SCAN AT (15 :16 );
        TEST DCV DCM VAL=5.0 TOL=5 DLY=5M [
                WRITE 'INCORRECT VOLTAGE SETTING OF +5V %NL%',
                BRANCH DDONE; ];

        SET SCAN AT (18:16);
        TEST DCV DCM VAL=-15 TOL=10 DLY=5M [



/* ATGMN419 TRAINGBD     31-OCT-83 11:48        PAGE 87.2 */

                WRITE 'INCORRECT VOLTAGE SETTING OF -15V %NL%',
                BRANCH DDONE; ];

        SET SCAN AT (17:16);
        TEST DCV DCM VAL=15 TOL=10 DLY=5M [
                WRITE 'INCORRECT VOLTAGE SETTING OF +15V %NL%',
                BRANCH DDONE; ];


                WRITE 'NOW TESTING OPAMP (U6)%NL%';

SETOPAMP:/*THE MUX CONNECTIONS FOR THE OPAMP TEST ARE:

 CHA = POSITIVE SIDE OF THE VOLTAGE SOURCE
 CHB = MINUS SIDE OF THE VOLTAGE SOURCE
 CHC = POSITIVE SIDE OF THE VOLTMETER
 CHD = NEGATIVE SIDE OF THE VOLTMETER
 REF 5 AND 6 CONNECT THE NEGATIVE SIDE OF THE VOLTAGE SOURCE TO CHB
*/

        SET MUX AT (6:10:4:5:5,6);
        USE FSUB=FAILCOMP;

/*THE VOLTAGE SOURCE IS CONNECTED TO THE INVERTING AND NONINVERTING
 INPUTS OF THE OPAMP. THE VOLTMETER IS CONNECTED BETWEEN THE OUT-
 PUT OF THE OPAMP AND THE NEGATIVE SUPPLY. THIS IS A "POWER ON"
 TEST. THEREFORE, THE POWER SUPPLY STATEMENTS MUST BE EXECUTED
 BEFORE THE TESTS ARE PERFORMED. A SMALL POSITIVE DIFFERENTIAL
 VOLTAGE IS APPLIED TO THE INPUTS. THIS WILL SEND THE OUTPUT OF
 THE OPAMP TO THE NEGATIVE RAIL VOLTAGE. THE VOLTMETER SHOULD READ
 APPROXIMATELY 0 VOLTS. NOW, A SMALL NEGATIVE VOLTAGE IS APPLIED
 TO THE INPUTS AND THE OUTPUT WILL GO TO THE POSITIVE RAIL VOLTAGE.
 THE VOLTMETER SHOULD NOW MEASURE THE TOTAL VOLTAGE SWING OF THE
 OPAMP. FOR EXAMPLE, WITH +15 VOLT AND -15 VOLT SUPPLIES, THE
 METER WILL MEASURE 30 VOLTS. IF THE OPAMP IS OPERATING WITH A
 SINGLE POWER SUPPLY, ONLY ONE RAIL MEASUREMENT WILL BE MADE.*/
```

```
/* ATGMN419 TRAINGBD     31-OCT-83 11:48        PAGE 102.1 */


/* MODIFIED OP-AMP TEST FOR DEVICE U6 WITH BRANCHING STATEMENT THAT
 ABORTS THE DAC (U5) TEST IF ITS OUTPUT DEVICE (U6) IS BAD */

U6: USE PMSG='DUAL LM1458';
         SET SCAN AT (117:118:181:18);
         SET DCV DCS V=100M I=20M DLY=0;
         TEST DCV DCS DMSG='UNABLE TO APPLY POSITIVE INPUT VOLTAGE' FAIL(
        8)
                [BRANCH U6E; ];
         MEAS DCV DCM HI=3 LO=-3 MAX=1
                DMSG='NEGATIVE RAIL' FAIL(8) [BRANCH U6E;];
         SET DCV DCS V=-100M I=20M DLY=0;
         TEST DCV DCS DMSG='UNABLE TO APPLY NEGATIVE INPUT VOLTAGE' FAIL(
        8)
                [BRANCH U6E; ];
         MEAS DCV DCM HI=33 LO=20 DLY=5M DMSG='POSITIVE RAIL' FAIL(8)
         [BRANCH U6E; ];
U6E:    SET DCV DCS V=0 I=20M;
         IF FAIL(8) THEN [WRITE ID=MESFILE'
        OUTPUT DEVICE FOR DAC U5 FAILED, ABORTING FURTHER TEST OF THIS%NL%';

        WRITE ID=MESFILE'
        DEVICE AND ALSO ABORTING TEST OF DAC U5%NL%';
         BRANCH U7; ];

U6_1:
         SET SCAN AT (77:181:21:18);
         SET DCV DCS V=100M I=20M DLY=0;
         TEST DCV DCS DMSG='UNABLE TO APPLY POSITIVE INPUT VOLTAGE'
                [BRANCH U6E; ];
         MEAS DCV DCM HI=3 LO=-3 MAX=1
                DMSG='NEGATIVE RAIL' [];
         SET DCV DCS V=-100M I=20M DLY=0;
         TEST DCV DCS DMSG='UNABLE TO APPLY NEGATIVE INPUT VOLTAGE'
                [BRANCH U6E; ];
         MEAS DCV DCM HI=33 LO=20 MAX=30 DMSG='POSITIVE RAIL'
                [];
U6_1E:   SET DCV DCS V=0 I=20M;
         CLEAR SCAN;     CLEAR MUX;

/* END OF MODIFIED OP-AMP TEST */



/* ATGMN419 TRAINGBD     31-OCT-83 11:48        PAGE 200.1 */

ADONE:  USE NOFSUB PMSG='' DMSG='' CMSG='';
        CLEAR SCAN;
        CLEAR MUX;
        /* DIGITAL TESTS INSERTED HERE */

DONE:   /* END OF DIGITAL TESTS */



/* ATGMN419 TRAINGBD     31-OCT-83 11:48        PAGE 240.1 */

        IF FAIL(1) THEN BRANCH FDONE;
        IF FAIL(2) THEN BRANCH FDONE;
        IF FAIL(3) THEN BRANCH FDONE;
        BRANCH PDONE;
FDONE:  LET DSTRG=DATE();
        WRITE ID=MESFILE
        '%036% %NL%%S% %NL%%035%
'       DSTRG;
        WRITE ID=MESFILE
                '%036% BOARD TRAINGBD FAILED%035%


'       ;
PDONE:  END;
```

Table 7-4
TEST PROCEDURE STREAM NUMBERS

Number	Test Description
%1S	Declarations
%2S	Insert SET SCAN statement for any device that could not be tested and which could not provide a FORM test
%3S	Set-up for opens and shorts tests
%4S	Continuity tests for open jumpers, normally open contacts, multi nailed nodes, normally closed contacts, jumpers, fuses, and inductors
%5S	Opens tests for open jumpers and normally open contacts
%10S	MUX and set-up for adjusting potentiometers
%11S	Adjust potentiometers
%14S	MUX and set-up for adjusting variable resistors
%15S	Adjust and test variable resistors at 0.2Vdc
%16S	Adjust and test variable resistors at 1.0Vdc
%17S	Format for adjusting variable resistors
%19S	Power and ground short test
%20S	Board shorts test
%24S	MUX and set-up for measuring resistance
%26S	200 mV guarded resistance tests
%27S	200 mV unguarded resistance tests
%28S	1 V guarded resistance tests
%29S	1 V unguarded resistance tests
%30S	5 V guarded resistance tests
%31S	5 V unguarded resistance tests
%32S	All resistor format tests
%33S	All potentiometer format tests
%40S	MUX and set-up for using ACZ
%42S	200 mV guarded ACZ tests
%43S	200 mV unguarded ACZ tests
%44S	1 V guarded ACZ tests
%45S	1 V unguarded ACZ tests
%46S	All ACZ format tests
%50S	MUX and set-up for diode tests
%51S	Forward diode characteristic tests
%52S	Forward diode format tests
%61S	Reverse diode characteristic tests
%62S	Reverse diode format tests
%65S	Nonlinear diode test
%70S	MUX and set-up for transistor gain tests
%71S	All transistor gain tests
%72S	Transistor format tests
%75S	Reserved for JFET applications
%76S	Reserved for JFET applications
%77S	Reserved for JFET applications
%78S	JFET format test set-up
%79S	JFET format tests
%80S	MUX and set-up for SCR I measure technique
%81S	SCR I measure tests
%82S	MUX and set-up for SCR V measure technique
%83S	SCR V measure tests
%84S	SCR format tests
%85S	MUX and set-up for zener tests
%86S	Guarded and unguarded zener tests

Table 7-4 (Continued)

Number	Test Description
%87S	Zener format tests
%95S	ASSIGN LGC and MUX for resistors w/digital stimulus
%97S	Resistor tests using digital stimulus and analog measurement
%100S	Power-on for OpAmp RAIL tests
%101S	MUX and set-up for OpAmp RAIL tests
%102S	All OpAmp tests
%110S	Power off for OpAmp RAIL tests
%200S	Boilerplate for digital test insertion
%240S	Close-out of test program
%250S	All subroutines

7.6 RUN DIGITAL TEST GENERATION

The running of digital ATG is independent of analog ATG and therefore the order in which the two are done is not important. If both are selected, however, the analog will be done first. If the User Digital Library file is to be used, its filespec must be stated in the "DTL =" option field.

The Digital Test Library contains only test language statements unique to the development of test procedures (bursts) for specific ICs. There are a number of other statements, such as declares and instrumentation statements, that must be included in the Test Program (.TPG or .TPX). Some of these statements are general to any board, be it all analog, all digital, or hybrid. Many of these general statements originate in the Analog Test Library or in the Analog ATG Task. Therefore, the Analog ATG must be run when generating any test program. Examples of these functions are shown in Table 7-5. Refer to the 227X Test Programming manual for an explanation of the individual statements.

The DTG task generates the Digital Test Program (.DTP) and a digital message file (.MSG). The .MSG should be reviewed for any problems that occurred during DTG. An example of a problem that would be found is that a test could not be found in any library for the component. A common reason is that the name provieded in the .CKT did not match the library name as shown in Appendix A. Another problem that would be discovered by reviewing the .MSG is a problem where a circuit condition did not allow all pins of a device to be disabled. The user could then analyze the problem and make modifications.

Table 7-5

GENERAL STATEMENTS INSERTED BY ATG INTO THE TEST PROGRAM (.TPG)

```
DECLARE statements
Connectivity Tests

ADONE:  USE NOFSUB;
        CLEAR SCAN;
        CLEAR MUX;

ON ABORT CALL PWRDN();
PWRUP:  SET PIO(0) HRLY(CLOSE 3);          /* GROUND PCB */
/* TYPICAL POWER SKELETON (Check for power sequencing requirements)

        SET PIO(0) HRLY(CLOSE 8); /* SYS ISOLATION RLY: UPS(4,5,6) */
        SET PIO(0) HRLY(CLOSE 7); /* FOR TPS IN SLOTS 1-3 */
        SET PIO(1) HRLY(CLOSE 1); /* USER ISOLATION RLY? */
/* 2275 Fixed users P.S. Control */
        SET PIO(0) HRLY(CLOSE 6); /* CONNECT TO DUT */
        DELAY 10M;
        SET PIO(0) RLY(CLOSE 12); /* TURN ON P.S. */
        DELAY 10M;
        SET DCV UPS(4) V=5 I=0.5 DLY=20M
                {WRITE 'UPS(4) OVERLOAD%NL%'; BRANCH DDONE;};
        SET TPS (1) VB DLY=50M;
        MEAS TPS (1) FAIL ()
                {WRITE 'TPS(1) OVERLOAD%NL%'; BRANCH DDONE;};
*/

ASSIGN LGC VIHA=4.5 VILA=0.0 VOHA=2.4 VOLA=0.8 LVLA(1 - 75);

DDONE:
        CALL PWRDN();

        /* DIGITAL TESTS INSERTED HERE */
DONE:   /* END OF DIGITAL TESTS */

        IF FAIL(1) THEN BRANCH FDONE;
        IF FAIL(2) THEN BRANCH FDONE;
        IF FAIL(3) THEN BRANCH FDONE;
        BRANCH PDONE;
FDONE:  LET DSTRG=DATE();
        WRITE ID=MSFILE
          '%036% %NL%%S% %NL%%035%
'       DSTRG;
        WRITE ID=MESFILE  '%036% BOARD LEXAR FAILED%035%
'       ;
PDONE:  END;

Required Subroutines
```

The complete TRAINGBD.MSG and TRAINGBD.DTP follow, with comments inserted beside them.

TRAINGBD.MSG

```
ATG Report On Digital Test Section SC:TRAINGBD.DTP 31-OCT-83 12:11
(DTGMN482) Libraries:   LB1:[2,11]DIGLIB740.DTL
----------------------------------------------------------------------
BOARD STATISTICS SUMMARY

        Device count              51
        Digital IC count          13
        Undrivable ICs          .  0
Target tester: 2275          Adaptor specified for: 2275


  Begin Database Creation

   7 Digital ATG Option commands found in SC:TRAINGBD.ATX
PROBLEM: No test program found for IC U5(NE5118).
PROBLEM: No test program found for IC U6(LM1458).
PROBLEM: No test program found for IC Y1(XTL)
PROBLEM: No test program found for IC Z1(RPAK8).
Lowest nail used    13
Highest nail used 4018


Logic Level Assignments for Nails
         LVLA(13-20,49-52,78,82-85,109-116,141-148,173-180,205-212,239
          -242,244,269-273,301-308,4001,4002,4009,4010,4017,4018)
```

Comments

The top of the .MSG identifies the report, the .DTP file, the date and time, and the libraries used to generate the test program. Next is a summary of statistics that provides data as follows:

Number of devcies on the board.

Number of digital ICs on the board.

Number of digital ICs which have inputs connected to a node which has been defined as undrivable by an UNDIV statement in the library header. This data is also in .ERP.

Target system specified on set-up page.

Fixture adaptor specified in .CKT Adaptor section.

Comments and problems found during the database creation phase of DTG. The items listed for TRAINGBD are a comment about the number of IC tested by ATO procedures and the problem message for each IC not tested by a library procedure. Each problem device is actually tested by one of the ATO procedures.

All nails that are assigned to each logic level as determined by ASSIGN LGC statements.

TRAINGBD.MSG (continued)

```
Results for Non-bussed ICs
--------------------------

         PIN RESULTS FOR U1 (7400)
                 Equivalent Set 1
          Pin  Type   Nail  Tested?  Comments
            1  IN      301    YES      A
            2  IN      302    YES      B
            3  OUT     303    YES      O
                 Equivalent Set 2
          Pin  Type   Nail  Tested?  Comments
            4  IN      303    YES      A
            5  IN      304    YES      B
            6  OUT     305    YES      O
         WARNING: DTG partially isolated U1(7400), eqv set # 2
         signal conflicts found near: Z1.3
                 Equivalent Set 3
          Pin  Type   Nail  Tested?  Comments
            8  OUT     110    YES      O
            9  IN      212    YES      A
           10  IN      111    YES      B
                 Equivalent Set 4
          Pin  Type   Nail  Tested?  Comments
           11  OUT     112    YES      O
           12  IN      110    YES      A
           13  IN      111    YES      B
     16 Test steps generated for U1(LB1:[2,11]DIGLIB740.DTL:7400).
                           (Extra text was included.) 1/0

         PIN RESULTS FOR U10 (7414)
                 Equivalent Set 1
          Pin  Type   Nail  Tested?  Comments
            1  IN      115    YES      IN
            2  OUT     114    YES      OUT
                 Equivalent Set 2
          Pin  Type   Nail  Tested?  Comments
            3  IN      114    YES      IN
            4  OUT     113    YES      OUT
                 Equivalent Set 3
          Pin  Type   Nail  Tested?  Comments
            5  IN      209    YES      IN
            6  OUT     111    YES      OUT
                 Equivalent Set 4
          Pin  Type   Nail  Tested?  Comments
            8  OUT      13    YES      OUT
            9  IN       78    YES      IN
                 Equivalent Set 5
          Pin  Type   Nail  Tested?  Comments
           10  OUT      19    YES      OUT
           11  IN       14    YES      IN
                 Equivalent Set 6
          Pin  Type   Nail  Tested?  Comments
           12  OUT     116    YES      OUT
           13  IN       20    YES      IN
     12 Test steps generated for U10(LB1:[2,11]DIGLIB740.DTL:7414).
                           (Extra text was included.) 2/0

         PIN RESULTS FOR U7 (NE555)
          Pin  Type   Nail  Tested?  Comments
            1  GND      16
            2  IN       84    YES      TRIG
            3  OUT      82    YES      OUT
            4  IN       17    HI       RST FORCED HI
            5  IN       83    NO       CV
            6  IN       84    HI       THR
            7  IN       85    NO       DIS
            8  VCC      18
      2 Test steps generated for U7(LB1:[2,11]DIGLIB740.DTL:NE555).
                           (Extra text was included.) 3/0

         PIN RESULTS FOR U8 (NE555)
          Pin  Type   Nail  Tested?  Comments
            1  GND      16
            2  IN       51    YES      TRIG
            3  OUT      49    YES      OUT
            4  IN       15    HI       RST FORCED HI
            5  IN       52    NO       CV
            6  IN       51    HI       THR
            7  IN       50    NO       DIS
            8  VCC      15
      2 Test steps generated for U8(LB1:[2,11]DIGLIB740.DTL:NE555).
                           (Extra text was included.) 4/0
```

Comments

The results of test generation for ICs are listed in the order that they appear in the test program. ICs not connected to any bus are first.

The results for each pin of a device are presented as follows:

Pin number.

Type of pin as defined in library header.

Fixture nail assigned to the node as defined in the .CKT.

Indication of being able to successfully test the pin.

Comments such as pin names defined in library.

Many ICs contain more than one equivalent, independent circuit. A common test procedure is repeated for each and the results are listed equivalent sets.

Following the pin results for each IC is a statement that indicates the number of test steps generated to test the device and from what library the test was taken. In the case of TRAINGBD, extra test steps were included for each IC by the INCLUDE ALLBURST [#IC(302) IL(302);] command in the ATO file. This fact is noted.

TRAINGBD.MSG (continued)

```
       PIN RESULTS FOR U9 (7474)
             Equivalent Set 1
        Pin  Type    Nail  Tested?  Comments
          1  IN       307    YES      CLR
          2  IN       306    NO       D
          3  IN       305    HI       CK
          4  IN       269    YES      PR
          5  OUT      308    YES      Q
          6  OUT      306    YES      QB
             Equivalent Set 2
        Pin  Type    Nail  Tested?  Comments
          8  OUT      271    YES      QB
          9  OUT      272    YES      Q
         10  IN       270    YES      PR
         11  IN       308    HI       CK
         12  IN       271    NO       D
         13  IN       273    YES      CLR
   18 Test steps generated for U9(LB1:[2,11]DIGLIB740.DTL:7474).  5/1

* Unbussed ICs Tested/Untested/ATO Replaced:  5/1


  Results for Bussed ICs
  ----------------------

          PIN RESULTS FOR U11 (2114)
           Pin  Type    Nail  Tested?  Comments
             1  IN       176    YES      A6
             2  IN       177    YES      A5
             3  IN       178    YES      A4
             4  IN       179    YES      A3
             5  IN       142    YES      A0
             6  IN       141    YES      A1
             7  IN       180    YES      A2
             8  IN       110    YES      CS
             9  GND       16
            10  IN       109    YES      WE
            11  TRI      206    YES      I4
            12  TRI      205    YES      I3
            13  TRI      244    YES      I2
            14  TRI      243    YES      I1
            15  IN       173    YES      A9
            16  IN       174    YES      A8
            17  IN       175    YES      A7
            18  VCC       15
      22 Test steps generated for U11(LB1:[2,11]DIGLIB740.DTL:2114).  6/1

          PIN RESULTS FOR U2 (Z80CPU)
           Pin  Type    Nail  Tested?  Comments
             1  TRI      274    YES
             2  TRI      275    YES
             3  TRI      276    YES
             4  TRI      237    YES
             5  TRI      238    YES
             6  IN       272    YES      CL
             7  TRI      243    YES      D4
             8  TRI      242    YES      D3
             9  TRI      244    YES      D5
            10  TRI      205    YES      D6
            11  VCC       15
            12  TRI      241    YES      D2
            13  TRI      206    YES      D7
            14  TRI      239    YES      D0
            15  TRI      240    YES      D1
            16  IN       210    YES      INT
            17  IN       211    YES      NMI
            18  OUT      207    YES      HALT
            19  TRI      209    YES      MREQ
            20  TRI      208    YES      IORQ
            21  TRI      144    YES      RD
            22  TRI      109    YES      WR
            23  OUT      145    YES      BUSAK
            24  IN       143    YES      WAIT
            25  IN       146    YES      BUSRQ
            26  IN       113    YES      RESET
            27  OUT      147    YES      M1
            28  OUT      148    YES      RFSH
            29  GND       16
            30  TRI      142    YES
            31  TRI      141    YES
            32  TRI      180    YES
            33  TRI      179    YES
            34  TRI      178    YES
            35  TRI      177    YES
            36  TRI      176    YES
            37  TRI      175    YES
            38  TRI      174    YES
            39  TRI      173    YES
            40  TRI      212    YES
          WARNING: DTG partially isolated U2(Z80CPU), eqv set # 1
          signal conflicts found near: Z1.4, Z1.5, Z1.7, Z1.8, Z1.9, U2.7,
               U2.8, U2.9, U2.10, U2.12
            and inability to use the nails: 146, 113
     163 Test steps generated for U2(LB1:[2,11]DIGLIB740.DTL:Z80CPU)  7/1
```

Comments

ICs that are connected to a bus cannot be tested until a bus test is done to determine if one IC on the bus is holding the bus node in one state. If the bus test fails, none of the bussed ICs will be tested. The bus test is done after the tests for non-bused ICs.

TRAINGBD.MSG (continued)

```
      PIN RESULTS FOR U3 (2114)
       Pin  Type    Nail  Tested?  Comments
         1  IN       176    YES      A6
         2  IN       177    YES      A5
         3  IN       178    YES      A4
         4  IN       179    YES      A3
         5  IN       142    YES      A0
         6  IN       141    YES      A1
         7  IN       180    YES      A2
         8  IN       110    YES      CS
         9  GND       16
        10  IN       109    YES      WE
        11  TRI      242    YES      I4
        12  TRI      241    YES      I3
        13  TRI      240    YES      I2
        14  TRI      239    YES      I1
        15  IN       173    YES      A9
        16  IN       174    YES      A8
        17  IN       175    YES      A7
        18  VCC       15
 22 Test steps generated for U3(LB1:[2,11]DIGLIB740.DTL:2114).  8/1

      PIN RESULTS FOR U4 (2716)
       Pin  Type    Nail  Tested?  Comments
         1  IN       175    YES      A7
         2  IN       176    YES      A6
         3  IN       177    YES      A5
         4  IN       178    YES      A4
         5  IN       179    YES      A3
         6  IN       180    YES      A2
         7  IN       141    YES      A1
         8  IN       142    YES      A0
         9  TRI      239    YES      D0
        10  TRI      240    YES      D1
        11  TRI      241    YES      D2
        12  GND       16
        13  TRI      242    YES      D3
        14  TRI      243    YES      D4
        15  TRI      244    YES      D5
        16  TRI      205    YES      D6
        17  TRI      206    YES      D7
        18  IN       112    LOW      CE
        19  IN       212    YES      A10
        20  IN       112    LOW      OE
        21  VCC       15
        22  IN       173    YES      A9
        23  IN       174    YES      A8
        24  VCC       15
      WARNING: DTG cannot isolate U4(2716), eqv set # 1
      signal conflicts found near: U2.37
        and inability to use the nails: 110
145 Test steps generated for U4(LB1:[2,11]DIGLIB740.DTL:2716).  9/1




 Summary of Digital Test Program SC:TRAINGBD.DTP


  **    9 ICs Tested
  **    0 ICs Untested

  **    1 ICs require additional bus disables.
          They are:      U2

  ** 4 Problems Detected

  ** User written text from SC:TRAINGBD.ATX was inserted.

         All BURSTs and BUSTESTs had user specified code included.
            9 ICs had user specified text included.


 * 8 of 8 bus nodes tested.
 BUS205  8       8   6   205,206,239-242,244,4009
         Bus pins BUSTESTed on U11 U2 U3 U4


 END of Digital Automatic Test Generation.
         0 library errors, 4 library problems.
```

Comments

A summary is included at the end of the .MSG, which includes the following information:

Number of ICs tested.

Number of ICs not tested.

Number of problems listed. In this case the problems refer to those listed at the beginning of the report indicating there was no library procedure for the devices.

An indication that something was inserted by the ATO file and that this included test steps for each IC and Bustest, as mentioned earlier.

An indication of success in generating bus tests.

TRAINGBD.DTP

```
/* DTGMN482 SC:TRAINGBD.DTP       31-OCT-83 12.11 */
/* Target tester. 2275            Adaptor specified for  2275 */


FUNDONE:          CALL PWRDN();

  WRITE 'NOW TURNING DOWN POWER FOR END OF FUNCTIONAL TESTING%NL%',


ADONE:   USE NOFSUB PMSG='' DMSG='' CMSG='';
         CLEAR SCAN;
         CLEAR MUX;

         WRITE 'NOW TURNING UP POWER FOR DIGITAL TESTING%NL%';

         ON ABORT CALL PWRDN();
PWRUP:   SET PIO(0) HRLY(CLOSE 3);        /* GROUND PCB */
         SET PIO(0) HRLY(CLOSE 6);        /* CONNECT FIXED P.S.  TO DUT */
         DELAY 10M;
         SET PIO(0) RLY(CLOSE 12);        /* TURN ON FIXED P.S.  OUTPUTS */
         DELAY 10M;

                  WRITE 'NOW TESTING NON BUSSED ICS%NL%';




  /* Un-bussed IC tests */

  U1:     BURST ; /* (7400) */


  $ IC(302) IL(302);


  /** TEST PROGRAM FOR IC U1 (7400)
  PIN  TYPE   NAIL  COMMENTS
    1  IN      301   A
    2  IN      302   B
    3  OUT     303   O
   **/
          IC(301,302) IL(302) IH(301) OS(303) OH(303);
          IH(302) OL(303);
          IL(301) OH(303);
          IL(302);
          ID(301,302) OI(303);
          $ ID(#) OI(#);


  $ IC(302) IL(302);

          /* U1 e1 used 301,302 */
          IC(301,302) IL(301,302),

  /** TEST PROGRAM FOR IC U1 (7400)
  PIN  TYPE   NAIL  COMMENTS

     4  IN      303   A
     5  IN      304   B
     6  OUT     305   O
    **/
           IC(303,304) IL(304) IH(303) OS(305) OH(305);
           IH(304) OL(305);
           IL(303) OH(305);
           IL(304);
           ID(303,304) OI(305),
           $ ID(#) OI(#);


   $ IC(302) IL(302);

           /* U10 e3 used 209 */
           /* Disable U2 */
           LOOP = 10 $ IC(113,272) IL(113,272);
           $ IH(272) END LOOP,
           $ ID(272);
           IC(209) IL(209);

   /** TEST PROGRAM FOR IC U1 (7400)
   PIN  TYPE   NAIL  COMMENTS
     8  OUT     110   O
     9  IN      212   A
    10  IN      111   B
    **/
           IC(111,212) IL(111) IH(212) OS(110) OH(110);
           IH(111) OL(110);
           IL(212) OH(110);
           IL(111);
           ID(111,212) OI(110);
           $ ID(#) OI(#);
```

Comments

The digital test program starts by identifying the DTG task name (DTGMN482), the program, the generation date, the target tester, and the fixture adaptor type for which test generation was done.

Functional testing of the op-amps and DACs required that power supplies be turned on. Power must be turned off after those tests and turned on again for digital testing. This FUNDONE routine, inserted by the REPLACE DIGINIT command in the .ATX file, serve both power functions.

The first tests inserted by DTG in the digital test program are for IC not on any bus. U1 is the first such IC. U1 has 4 equivalent sets. An independent section of the burst is inserted for each set. A command in the .ATX file (INSERT ALLBURST) caused the test step, $IC(302) IL(302); to be inserted at the beginning of each burst and each independent section. This step inhibits the on board clock for all testing.

TRAINGBD.DTP (continued)

```
$ IC(302) IL(302);

        /* U1 e3 used 212 */
        /* U10 e3 used 209 */
        IC(209,212) IL(209,212);

/** TEST PROGRAM FOR IC U1 (7400)
PIN  TYPE   NAIL  COMMENTS
 11  OUT     112   O
 12  IN      110   A
 13  IN      111   B
 **/
        IC(110,111) IL(111) IH(110) OS(112) OH(112);
        IH(111) OL(112);
        IL(110) OH(112),
        IL(111);
        ID(110,111) OI(112);
        $ ID(#) OI(#);
        END BURST;       /* U1 */
            |
            |
            |
            |
            |
            |
            |
            |

 /* Parallel bus tests   */
 /*   RATIO = 1.3 */

  ASSIGN LGC VCDH=3.5 VCDL=.2 VCST=1.5 VTT=1.5
          VIHA=4.5 VILA=0.0 VOHA=2.2 VOLA=1.4 LVLA(13-20,49-52,78,82-85,
           109-116,141-148,173-180,205-212,239-242,244,269-273,301-308,
           4001,4002,4009,4010,4017,4018);

 CLEAR MUX;
 CLEAR SCAN;

 BUS205: BUSTEST AT(205,206,239-242,244,4009) IST=10U,


 $ IC(302) IL(302);

        /* U11 used 110 */
        /* Disable U2 */
        LOOP = 10 IC(113,272) IL(113,272);
        IH(272) END LOOP;
        $ ID(272);
        /* U4 used 112 */
        /* U11 used 109 */
        /* Inhibit U2 */
        LOOP = 10 IC(113,272) IL(113,272);
        IH(272) END LOOP;
        $ ID(272);
        /* U4 used 141,142,173-180,212 */
        IC(109,110,112,141,142,173-180,212) IL(109,141,142,173-180,212)
           IH(110,112);
    BUSDATA;
        $ PD(205,206,239-242,244,4009); $ ;
        OS(205,206,239-242,244,4009) OL(205,206,239-242,244,4009);
        $ PN(205,206,239-242,244,4009); $ OI(205,206,239-242,244,4009);
        $ PU(205,206,239-242,244,4009); $ ;
        OS(205,206,239-242,244,4009) OH(205,206,239-242,244,4009);
        $ PN(205,206,239-242,244,4009); $ OI(205,206,239-242,244,4009);
    HFORCE NAME='U4';
        /* U4 used 112,141,142,173-180,212 */
        IC(112,141,142,173-180,212) IL(112,141,142,173-180,212);
    LFORCE NAME='U4';
        /* U4 used 112,141,142,173-180,212 */
        IC(112,141,142,173-180,212) IL(112,141,142,173-180,212);
    HFORCE NAME='U2';
        IC(143,146,210,211) IH(143,146,210,211);
 /*RESET PROCESSOR*/
        $ IC(272) IL(272);
        $ IH(272);
        IC(113) IL(113);
        $ IL(272);
        $ IH(272);
        $ IL(272);
        $ IH(272);
        $ IL(272);
        $ IH(272);
        $ IL(272);
        $ IH(272);
        $ IL(272);
        $ IH(272),
        IH(113);
```

Comments

The remainder of the un-bussed IC tests are omitted.

Parallel bus testing is inserted next. The ASSIGN LGC statement sets levels for all bussed IC nodes. The BUSTEST statement selects all bus nodes. Only the first part of the bus test is shown.

TRAINGBD.DTP (continued)

Comments

```
                |
                |
                |
                |
                |
                |
                |

 ASSIGN LGC VCDH=3.5 VCDL=.2 VCST=1.5 VTT=1.5
          VIHA=4.5 VILA=0.0 VOHA=2.4 VOLA=0.8 LVLA(13-20,49-52,78,82-85,
           109-116,141-148,173-180,205-212,239-242,244,269-273,301-308,
           4001,4002,4009,4010,4017,4018);

/* Bussed IC tests */
```

Tests for bussed ICs are inserted next and would be executed if bus testing passed. The first part of the test for U11 is shown. The remainder of the DTP is omitted.

```
U11:     BURST ; /* (2114) */


$ IC(302) IL(302);

          /* U1 e3 used 111,212 */
          /* Disable U2 */
          LOOP = 10 $ IC(146,272) IL(146,272);
          $ IH(272) END LOOP,
          $ ID(272);
          /* U4 used 112 */
          IC(111,112,212) IL(111,212) IH(112);

/** TEST PROGRAM FOR IC U11 (2114)
PIN  TYPE   NAIL  COMMENTS
  1  IN      176   A6
  2  IN      177   A5
  3  IN      178   A4
  4  IN      179   A3
  5  IN      142   A0
  6  IN      141   A1
  7  IN      180   A2
  8  IN      110   CS
  9  GND      16
 10  IN      109   WE
 11  TRI     206   I4
 12  TRI     205   I3
 13  TRI     244   I2
 14  TRI    4009   I1
 15  IN      173   A9
 16  IN      174   A8
 17  IN      175   A7
 18  VCC      15
 **/
/* PAUSE 500USEC FOR CHARGE PUMP */
          LOOP = 75 ;
           END LOOP,
          IC(109,110,141,142,173-180,205,206,244,4009) IL(141,142,173-180,
            205,206,244,4009) IH(109,110);
          IL(110);
/*WRITE ALL ONE'S TO LOCATION ZERO AND READ IT BACK*/
/* WRITE CYCLE */
          IH(205,206,244,4009);
          IL(109);
          IH(109);
          IH(110);
/* READ CYCLE */
          ID(205,206,244,4009) IL(110) OS(205,206,244,4009) OH(205,206,244,
            4009);
          IH(110) OI(205,206,244,4009);
/*WRITE ZERO'S TO LOCATION ZERO AND READ IT BACK*/
                |
                |
                |
                |
                |
                |
```

7.7 MERGE ANALOG AND DIGITAL PROGRAMS

Once the analog and digital test programs have been generated and edited as required, the two programs are merged into a single program. The "Merge test programs" option selects this step.

7.8 USER MODIFICATIONS IN ATG PROGRAMS

The source test program (.TPG or .TPX) developed by ATG is coded in the high-level test language described in the Test Programming manual. The user should be familiar with this language because there are several modifications that a user may need to make to the .TPG. For instance, the digital logic levels may be different from the standard levels which ATG always uses. Another modification may be to insert a user-written test procedure for a device that is not considered in the various libraries. When analog or digital ATG is unsuccessful in generating a test for a device that has an applicable library routine, a FORM test, or a dummy test, is included in the program. If ATG cannot find an applicable routine in a library, nothing is put into the program. The first executable statement of the FORM test is a branch around the FORM test.

The FORM test becomes part of the test program by removing the branch statement from the test and by making any needed changes to the parameters in the test. This would be done by using the Debug procedures.

Since the nail assignment software uses the ATG-developed test program to assign nails, devices not contained in the test program will not be assigned test nails in the wiring list. Make sure, therefore, that either a real or a dummy test for each device is included in the test program.

ATG files that contain temporary nail numbers are so indicated by an "X" in the file extension:

ATG Files	Nail Assignment Output
.TPX	.TPG
.IDX	.IDD
.ATX	.ATO

When these files are updated by the nail assignment software, the file extension is automatically changed to the more familiar .TPG, .IDD, .ATO form, indicating that they contain actual nail numbers.

7.8.1 Required Instrumentation Modifications

General functions unique to digital testing originate in the DTG task. Some of these require editing by the user prior to running the translator. A discussion of each of the statements which requires editing is contained in the next three paragraphs. All general statements are listed in Table 7-5.

If any nails are assigned to test ECL devices, 'B' logic voltages will be assigned with the additional statements

```
VIHB=-0.9 VILB=-1.8 VOHB=-1.1 VOLB=-1.5 LVLB( <nail list> )
```

The ASSIGN LGC statement inserted into the test program by DTG contains typical logic level voltages for TTL and ECL devices. It also indicates that these logic levels are applied to all driver/sensor pins in the .CKT file. If the logic levels for the circuit are different from these typical values, the user must change the values entered by DTG to agree with the circuit being tested.

<u>Examples</u> showing both standard and user-modified ASSIGN LGC statements:

Standard statement inserted by DTG:

```
ASSIGN LGC VIHA=4.5.  VILA=0.0  VOHA=2.4 VOLA=0.8 LVLA (nail list)
```

Possible user-modified statement:

```
ASSIGN LGC VIHA=4.5 VILA=0   VOHA=2.4 VOLA=0.8 LVLA (30-150, 175, 248)
           VIHB=3   VILB=0   VOHB=1.8 VOLB=0.4 LVLB (15-26, 176, 201-220);
```

SET DCV UPS(4) V=5 I=0.5 DLY=20M; and SET TPS(1) VB DLY=50M; statements are inserted into the test program by DTG for the purpose of providing VCC.

It is important that the user consider these statements before actually using them in the test program. The group of power statements is, therefore, inserted as a comment (enclosed in /* */ symbols). If ATG has inserted the proper power statements, then the user need only remove the comment symbols. In many instances, the user may have already caused a proper power statement to be inserted from an .ATO file.

The user can select the configuration of power supplies to be built into the system. Thus, the configuration may not be the same as that assumed by DTG. The possible power-supply options could result in one or more of the following being specified in SET DCV UPS( ) or SET TPS( ) statements:

```
TPS(1)          TPS(4) UPS(4)
TPS(2)          TPS(5) UPS(5)
TPS(3)          TPS(6) UPS(6)
```

The VCC power selected by the SET DCV UPS( ) or SET TPS( ) statement is applied to the circuit board under test via a relay inside the relay box. The SET PIO( ) command exercises the relay driver which, in turn, energizes the relay. Since the relay drivers are a system option, their selection may vary from the configuration of the typical system, as assumed by DTG. Thus, the SET PIO( ) statement is also included between the same set of comment symbols as the SET DCV UPS( ) or SET TPS( ) statement. The possible relay driver options could result in one of the following being specified in the SET PIO( ) statement:

```
PIO(1)  HRLY(1)         Provided by DTG
PIO(1)  HRLY(2)
PIO(1)  HRLY(3) --
PIO(1)  HRLY(4)   ¦
PIO(1)  HRLY(5)   ¦__   If these are used, it is recommended that
PIO(1)  HRLY(6)   ¦     suppression diodes also be installed.
PIO(1)  HRLY(7)   ¦
PIO(1)  HRLY(8) --
```

<u>Example</u> showing both standard and user-modified power statements for a situation where the UPS is being used for -5 V and it is important that this supply be connected to the circuit before the TPS and disconnected from the circuit after the TPS.

Standard power statements inserted by DTG:

```
ON ABORT CALL PWRDN();
PWRUP:    SET PIO(0) HRLY(CLOSE 3);       /* GROUND PCB */
/* TYPICAL POWER SKELETON (Check for power sequencing requirements)

          SET PIO(0) HRLY(CLOSE 8); /* SYS ISOLATION RLY: UPS(4,5,6) */
          SET PIO(0) HRLY(CLOSE 7); /* FOR TPS IN SLOTS 1-3 */
          SET PIO(1) HRLY(CLOSE 1); /* USER ISOLATION RLY? */
/* 2275 Fixed users P.S. Control */
          SET PIO(0) HRLY(CLOSE 6); /* CONNECT TO DUT */
          DELAY 10M;
          SET PIO(0) RLY(CLOSE 12); /* TURN ON P.S. */
          DELAY 10M;
          SET DCV UPS(4) V=5 I=0.5 DLY=20M
                  [WRITE 'UPS(4) OVERLOAD%NL%'; BRANCH DDONE;];
          SET TPS (1) VB DLY=50M;
          MEAS TPS (1) FAIL ()
                  [WRITE 'TPS(1) OVERLOAD%NL%'; BRANCH DDONE;];
*/

ASSIGN LGC VIHA=4.5 VILA=0.0 VOHA=2.4 VOLA=0.8 LVLA(12,13,34-38);
/* Un-bussed IC tests */
                  ¦
                  ¦
          digital test statements
                  ¦
                  ¦
DDONE:
                  CALL PWRDN();
                  ¦
                  ¦
          Ending statements
                  ¦
                  ¦
SUBROUTINE PWRDN();    /* POWER DOWN ROUTINE */
PWRD: /*
          SET UPS(4) V=0.0 I=0.2 FAIL();
UPSL4:    TEST UPS(4) HI=0.05 FAIL()[BRANCH UPSL4]; /* WAIT FOR CAPS */
          CLEAR UPS(4);   /* DISCONNECT POWER SUPPLY */
          CLEAR TPS;
          SET PIO(0) RLY(OPEN 12); /* TURN OFF 2275 FIXED P.S. */
          DELAY 50M;
          SET PIO(10 HRLY(OPEN 1); /* USER ISOLATION RLY? */
          SET PIO(0) HRLY(OPEN 8); /* SYS ISOLATION RLY: UPS(4,5,6) */
          SET PIO(0) HRLY(OPEN 7); /* ISOLATE TPS IN SLOTS 1-3 */
*/
          SET PIO(0) HRLY(OPEN 6); /* DISCONNECT 2275 P.S. FROM DUT */
          SET PIO(0) HRLY(OPEN 3); /* GND RLY */
END SUBROUTINE PWRDN:
```

The statement ON ABORT CALL PWRDN(); inserted at the beginning of the test program is meant to bring all supply voltages to zero before relays are opened or closed. Users of the TPS should add either a loop, using a DC measure statement, that terminates only when the TPS voltage has fallen below 100 mV or a fixed delay based on a calculation of RC constants to ensure that the TPS and on-board capacitors have discharged to below 100 mV. Failing to provide a proper power-down sequence for the TPS can cause serious damage to multiplexer relays.

User-modified statements are required to change the UPS voltage to -5 V and to reverse the disconnect sequence in the PWRDN subroutine. Only the statements that changed are shown.

```
TYPICAL POWER SKELETON (Check for power sequencing requirements)
SET DCV UPS(4) V=-5 I=0.5 DLY=20M;

PWRD:
        CLEAR TPS;
        SET UPS(4) V=0 I=0.5;
            |
            |
```

The BURST and the END BURST statements are inserted by DTG at the beginning and the ending of each test procedure that is generated. An exception exists when more than one burst is required for the test procedure. Then the BURST and END BURST statements must be included in the library procedure and the DTG inserts are deleted. DTG adds the label to all BURST statements whether inserted by DTG or by the library routine. When multiple bursts exist, the label includes the burst count after the IC name for all BURST statements after the first one.

When DTG is not successful in generating a test for an IC or an independent section of an IC, it writes a message that notes this fact and shows a skeleton of the test burst. The skeleton burst has a branch statement preceding it. This makes it easy for the user to design a test burst for the device and to edit the program to include it. The skeleton burst statements written by DTG would be included for each independent section that failed. After the burst has been edited to make it usable, the branch statement should be deleted so that the burst will be executed.

7.8.2 Methods of Modifying Programs

When the initial analysis for testing the board was done, it may have been determined that a certain modification to a normally generated test program is needed. In this case an .ATX or .ATO file would have been created so that ATG could automatically include the modification as ATG is being run. Refer to Section 5 for information about writing .ATX or .ATO files.

If the need for a modification is not determined until after ATG is run, the .TPX or .TPG file could be edited using the EDIT mode. If this edit causes changes to any node connections, there could be problems with diagnostic messages when the program is run because of incomplete or incorrect information in the .IDD file. THis method also leaves the document set without source information so that a latter rerun of ATG would not contain the edited items. It may be more desireable to create an .ATO file and rerun .ATG.

7.8.3 Other Alternatives in Correcting Programs

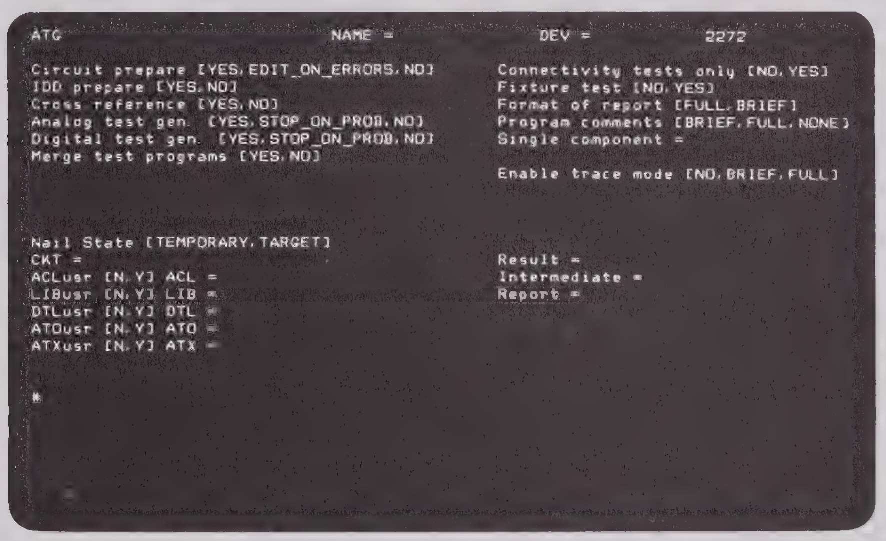
The analysis of untested components resulting from ATG may determine that a change in the circuit description, such as a wider error limit flagspec, will help improve the test strategies of the board. To make a change in the circuit description, the latest version of the .CKT file is brought into the Edit mode. After appropriate changes have been made, the edited version of the .CKT file is resubmitted to the circuit compiler. Refer to Section 6 for .CKT preparation information.

The analysis of untested components may determine that a change to the test library would be appropriate because there will be many boards with similar circuit configurations. A library change could be made that might improve the ability of ATG to generate a test. After making changes in the library, recompile the library source file. Refer to the 227x Test Library Programming manual for library routine preparation information.

7.9 AUTOMATIC TEST GENERATOR (ATG) MODE

The ATG monitor page allows the user to access the tasks required to create the test program (.TPG or .TPX) from the circuit description (.CKT).

Figure 7-5 and the following option descriptions provide information about each option and how to use it.



```
ATG                               NAME =                    DEV =          2272

Circuit prepare [YES,EDIT_ON_ERRORS,NO]       Connectivity tests only [NO,YES]
IDD prepare [YES,NO]                          Fixture test [NO,YES]
Cross reference [YES,NO]                      Format of report [FULL,BRIEF]
Analog test gen.  [YES,STOP_ON_PROB,NO]       Program comments [BRIEF,FULL,NONE]
Digital test gen.  [YES,STOP_ON_PROB,NO]      Single component =
Merge test programs [YES,NO]
                                              Enable trace mode [NO,BRIEF,FULL]



Nail State [TEMPORARY,TARGET]
CKT =                                         Result =
ACLusr [N,Y] ACL =                            Intermediate =
LIBusr [N,Y] LIB =                            Report =
DTLusr [N,Y] DTL =
ATOusr [N,Y] ATO =
ATXusr [N,Y] ATX =


*
```

Figure 7-5. ATG Monitor Options Page.

Circuit prepare [YES,EDIT_ON_ERRORS,NO]

[YES] YES selects the CDL compile phase of ATG which compiles the .CKT files into the .WOR file. Standard CDL compiler messages will be displayed and the .ERR file made. When errors are encountered, the compile will stop after the 1st, 2nd or 3rd pass, as applicable.

[EDIT_ON_ERRORS] Causes the monitor to go into edit mode automatically upon completion of the current pass (either 1 or 2) of the circuit compiler if there were any errors detected. The file that is being edited contains all error messages generated by the compiler. When the user exits the editor, the corrected file is automatically saved on the source device, with the next higher version, and the compiler is automatically invoked again. This process is repeated until no errors occur. At this time the correct circuit listing will become the latest source file. It is not necessary to delete the error messages from any of the listings with errors because the compiler ignores these messages and does not pass them to the output. While editing the file, the user may exit the edit loop prior to obtaining an error-free compile by pressing ESC E on the keyboard. If the TECO editor is being used, entering the TECO command

```
EC $$
```

before pressing ESC E will close the edited file and save it, before control is returned to the ATG page.

[NO] Selected when the .WOR file already exists and the ATG process can bypass the CDL compile task.

IDD prepare [YES,NO]
Specifies whether or not the In-circuit Diagnostic Data file and digital interconnect file for the UUT are to be prepared by ATG.

[YES] The default for this option, YES directs ATG to create the .IDD and .DAI files for the UUT.

[NO] If the .IDD and .DAI files already exist, selecting NO informs ATG not to create these files.

Cross reference [YES,NO]
A good method of checking a new circuit description file is to observe the cross reference report which is generated by ATG when requested.

[YES] A complete report can be generated by specifying YES for this option. When this is done the ATG process generates the report and then continues with other ATG tasks. The cross reference report (.AXR) contains all component values, connections, and flagspecs. It also lists components with missing or unusual values and lists nodes with no connections.

[NO] When NO is enabled, the ATG does not attempt to compile a cross reference report.

Analog test gen. [YES,STOP_ON_PROB,NO]
Analog test generation must be enabled to produce the analog test needed before digital testing can be done. Therefore, analog test generation must be run either before digital test generation or at the same time.

[YES] YES selects the task phase of ATG which develops the analog portion of the test program (.ATP). The .WOR file must have been previously compiled or "Circuit prepare" must also be specified.

[STOP_ON_PROB] Selecting STOP ON PROB directs ATG to stop analog test generation if problems are detected in the .WOR, .LIB, or .ATO files.

[NO] Selected when the .ATP file already exists or when only other ATG tasks are to be run.

Digital test gen. [YES,STOP_ON_PROB,NO]

Digital test generation writes the tests (called digital "bursts") for digital ICs on the UUT.

[YES] YES selects the DTG phase of ATG which develops the digital portion of the test program (.DTP). The .IDD, and .DAI files must have been previously compiled or "Circuit prepare" and "IDD prepare" must also be enabled.

[STOP_ON_PROB] Specifying STOP ON PROB allows ATG to stop digital test generation if problems are detected in the .IDD, .IDX, .DAI, .DTL, .ATO or .ATX files.

[NO] Selected when the .DTP file already exists or when only other ATG tasks are to be run.

Merge test programs [YES,NO]

Analog test generation and digital test generation write separate test programs for each class of component, analog or digital, on the UUT. To produce a complete test program for the board, these two test files must be merged.

[YES] YES selects the phase of ATG which merges the analog and digital result files (.ATP and .DTP) into a single test program file (.TPG or .TPX).

[NO] Selected when a full test program is not desired.

Resuming test gen. [NO,YES]

The analog and digital phases of the ATG process include a means of saving their status. If the system has been intentionally halted, it can be restarted at the point indicated by the last status save. A halt may result from program control or operator control (pressing CTRL C). After there has been a halt and the system is ready to continue, the ATG page will appear on the display.

[NO] Selected when ATG is to be run on a new circuit description file or a new test generation run is to be made with an existing (possibly edited) file.

[YES] In addition to selecting YES for this option, the following other preparations must be made to continue the ATG process where it was halted:

a. Select the ATG phase where the halt occurred (Analog or Digital).
b. Select NO for the "Circuit prepare" option.
c. Select NO for the ".IDD prepare" option.

Nail state [TEMPORARY,TARGET]

Option does not appear when the target system is a 2270.

[TEMPORARY] indicates that the .CKT file contains temporary nail numbers that the nail assignment routine will convert to actual numbers. The resulting In-Circuit Diagnostic Data file and Test Program file will be assigned extensions .IDX, and .TPX, respectively, to indicate that they also contain temporary nail numbers.

[TARGET] indicates that the .CKT file contains actual nail numbers, and the resulting ATG files have extensions .IDD and .TPG.

Connectivity tests only [NO,YES]

Enabling this option can prepare a Connectivity Test Program (.CTP) for shorts and opens on the pc board without a complete test program being produced.

[NO] No .CTP file is produced when the default setting, NO, is enabled. The shorts tests will be included as the first tests in the Analog Test Program (.ATP).

[YES] Enabling YES directs ATG to create an independent Connectivity Test Program (.CTP) that includes complete shorts and opens tests for the UUT. This test is done with a circuit board (with or without components) mounted on the fixture. A shorts check is performed between all pairs of nails, except where nails are connected to the same node on the circuit. An opens test is done to check for continuity between nails on the same node.

Fixture Test [NO,YES]

Enabling this option can produce a complete independent Fixture Test Program (.FTP) to test continuity to the test fixture.

[NO] ATG does not write a Fixture Test Program (.FTP) when NO is selected.

[YES] Specifying YES directs ATG to provide a Fixture Test Program (.FTP) to test for complete continuity on a test fixture. An opens test is done with a shorting plate mounted on the fixture. The plate is removed for the shorts test. A continuity check is performed from every nail to a common point.

Format of report [FULL,BRIEF]

Specifies the extent of data to be included in the CDL compilation report (.ERR), and the digital ATG report (.MSG). When the "Circuit prepare EDIT ON ERRORS" option is enabled, this option is ignored and a FULL report is automatically provided for editing. The .RPT is not affected by this option.

[FULL] When FULL is selected, the .ERR file contains the complete circuit description file, including errors. The .MSG file contains the number of test steps generated for each IC, any problems encountered with DISable or INHibit sections, board statistics, and IC pin reports.

[BRIEF] Specifying BRIEF results in only the errors being written to the .ERR file, and omits the IC pin reports from the .MSG file.

Program comments [BRIEF,FULL,NONE]

Indicates how explicit ATG will be in explaining its actions through comments in the resulting test program.

[BRIEF] Enabling BRIEF results in no comments being included in the Analog Test Program and numbers only instead of full page headers. Comments within .REPEATs in the Digital Test Program are not included.

[FULL] Selecting FULL includes all test report comments and full page headers in the Analog Test Program. The Digital Test Program includes comments within .REPEATs for each loop and comments defining DISabled or INHibited IC pin types and nails.

[NONE] No comments will be included in the ATG produced test program.

Single component =

A component name causes ATG to develop a test program for that analog or digital component only. The component name must be the same as given in the circuit description. Entering an asterisk (*) followed by a component type directs ATG to develop a test program for all components of that type.

Enable trace mode [NO,BRIEF,FULL]

The amount of information displayed on the screen during test generation is controlled by this option. When ATG is run from batch, this option also specifies how much information is recorded in the batch listing file (.BLI).

[NO] When NO is specified, the trace feature is minimal. It provides the information which tells the operator what portion of ATG is presently being processed and provides the data needed for the report file.

[BRIEF] When BRIEF is specified, the trace provides a little more data about what was done in each part of ATG and tells the operator in a little more detail what portion of ATG is being processed.

[FULL] When FULL is specified, the trace will list each library statement that is processed. It will also provide each of the values calculated by the circuit evaluator routine in the ATG process.

If BRIEF is specified and a Test Selection Language error is detected, the trace mode will automatically be considered to be FULL. When Debugging is completed, it is recommended that the trace mode be reset to NO so that the next user will not waste a considerable amount of time because the "Enable trace mode" option was left in FULL.

When a FULL trace is specified by the "Enable trace mode" option, there is a very large amount of data being transferred. The video display is effective as a trace file device because the speed of the processing can be controlled by the CONTROL S and CONTROL Q keys on the keyboard. Running ATG from a batch file writes the trace data to the batch listing (.BLI) file as well as to the screen. Space on a disk device will be filled up rapidly when FULL is selected; therefore, always purge the .BLI files upon completion of the debugging. A trace can be turned on while ATG is running by pressing CTRL T.

CKT =

If not specified in this option, the device and name for the circuit description file default to the "NAME =" and "DEV =" option-values.

ACLusr [N,Y] ACL =

Selecting N indicates there is no user-created analog components library and only the update and system libraries are used. Enabling Y indicates the existence of a user library and the "ACL =" option appears. If no alternate file specification is entered here, the "DEV =" and "NAME =" option-values are assumed. This option applies only to the "Circuit prepare" function.

LIBusr [N,Y] LIB =

The existence of a user-created analog test library is signalled by this option. Select N if there is no user library. When Y is selected, the "LIB =" option appears. Enter the device and library name of user-created analog test library if they differ from the "DEV =" and "NAME =" option-values. When Y is enabled, ATG will look in the user library first for an OUTLINE, DIRECTORY, or TEST when writing analog tests.

DTLusr [N,Y] DTL =
Setting this option to N indicates there is no user-created digital test library. Enabling Y causes the "DTL =" option to appear. Enter the device and library name of user-created digital test library if they differ from the "DEV =" and "NAME =" option-values. When a file is specified; DTG will look in this file first when generating digital tests.

ATOusr [N,Y] ATO =
or
ATXusr [N,Y] ATX =
Specify N for this option when a user test options file is not to be used for the ATG
run. Selecting Y causes ATG to look for a test option file. The "ATO =" or "ATX =" option appears. Enter the device and file-name of the options file. If no device or name is entered, the system will use the "DEV =" and "NAME =" option-values. When Y is selected, ATG will automatically perform the edits that are in the named procedure.

Result =
Indicates to ATG the file device. The ASCII character result files of concern to the user are:

Full Test Program (.TPG or .TPX)
Analog Test Program (.ATP)
Digital Test Program (.DTP)
Fixture Test Program (.FTP)
Connectivity Test Program (.CTP)

If no entry is made, the file device will assume the device entered at the top of the options page.

The above result files can be viewed using the Editor, Leditor, Type or Print utility.

The RESult device also contains the non-observable In-Circuit Diagnostic Data (.IDD) file which is used by both ATG and the Run-Time System.

Intermediate =
Name of file device where all temporary files, generated by ATG, are stored. These are files which are a necessary part of ATG, but are not provided in a user readable form. They have no value once the ATG process is completed.

Report =
Indicates to ATG the file device for report listings produced by ATG. If no device is specified, the device in the "DEV =" option is used. The file device may not be TI: or LP:. Report listing must be created on disk and then viewed or printed with the Type or Print utility. The report listings and appropriate extensions are:

Circuit Description Compiler Error Listing (.ERR)
Digital Pre-processor Report (.ERP)
Digital ATG Report (.MSG)
Analog ATG Report (.RPT)
Cross Reference Report (.AXR)

Nail Assignment – Section 8

8.1 INTRODUCTION

For the purpose of describing the nail assignment software, GenRad's circuit board test systems can be grouped into 3 categories:

> Systems that have a dedicated driver/sensor connected to each digital (or hybrid) test nail (i.e., 2270 systems).
>
> Multiplexed (MUXed) systems in which each driver/sensor is shared by a group of test nails (i.e., 2271, 2272 and 2275 systems).
>
> Systems that have the SpeedPlus option, a feature that provides the system with additional types of special test nails.

The 2270 system fits in the first category. It does not require the use of nail assignment software, since each digital (or hybrid) test nail has its own driver/sensor connected to it. As a convenience to the user, however, a subset of the nail assignment software is available to develop a 2270 fixture wiring list, via the Fixture Report monitor mode (para 8.9).

In the second category are the 2271, 2272 and 2275 systems with multiplexed driver/sensors. In the 2271 and 2275, each group of 16 test nails shares 4 driver/sensors; in the 2272, each group of 16 nails shares 2 driver/sensors. This means that digital nodes which must be tested concurrently cannot be connected to more than 4 (in the case of the 2271 or 2275) or 2 (in the case of the 2272) test nails from each multiplexed group of 16 (i.e., 1-16, 17-32, etc). To ensure that these system considerations are adhered to, the user is urged to let the nail assignment software, described in this section, assign the test nails.

The third category consists of systems that use the SpeedPlus option. In addition to having standard hybrid test nails, systems with this feature have the following additional nail types: fast, clock drive, clock sync and trigger. Nail assignment should be used to ensure that these nails are also properly assigned.

The general operation of nail assignment is shown in Figure 8-1.

Before using nail assignment, the user must first prepare the circuit description (.CKT) file and the test program (.TPX) [1] file. Since the actual test nails (also referred to as target nails) are not known at this time, the user inserts temporary nail numbers in the ADAPTOR section (i.e., %ADAPTOR:227x TEMP) of the .CKT file. CKTGEN can be used to prepare the circuit description and ATG is used to compile the .CKT file and develop the test program.

---

[1] An X is used in certain file extensions (.TPX, .ATX, .IDX) to indicate that the file contains temporary nail numbers. When nail assignment replaces the temporary nail numbers with target nail numbers, it automatically renames the files .TPG, .ATO, .IDD.

Nail Assignment – Section 8

8.1 INTRODUCTION

For the purpose of describing the nail assignment software, GenRad's circuit board test systems can be grouped into 3 categories:

> Systems that have a dedicated driver/sensor connected to each digital (or hybrid) test nail (i.e., 2270 systems).
>
> Multiplexed (MUXed) systems in which each driver/sensor is shared by a group of test nails (i.e., 2271, 2272 and 2275 systems).
>
> Systems that have the SpeedPlus option, a feature that provides the system with additional types of special test nails.

The 2270 system fits in the first category. It does not require the use of nail assignment software, since each digital (or hybrid) test nail has its own driver/sensor connected to it. As a convenience to the user, however, a subset of the nail assignment software is available to develop a 2270 fixture wiring list, via the Fixture Report monitor mode (para 8.9).

In the second category are the 2271, 2272 and 2275 systems with multiplexed driver/sensors. In the 2271 and 2275, each group of 16 test nails shares 4 driver/sensors; in the 2272, each group of 16 nails shares 2 driver/sensors. This means that digital nodes which must be tested concurrently cannot be connected to more than 4 (in the case of the 2271 or 2275) or 2 (in the case of the 2272) test nails from each multiplexed group of 16 (i.e., 1-16, 17-32, etc). To ensure that these system considerations are adhered to, the user is urged to let the nail assignment software, described in this section, assign the test nails.

The third category consists of systems that use the SpeedPlus option. In addition to having standard hybrid test nails, systems with this feature have the following additional nail types: fast, clock drive, clock sync and trigger. Nail assignment should be used to ensure that these nails are also properly assigned.

The general operation of nail assignment is shown in Figure 8-1.

Before using nail assignment, the user must first prepare the circuit description (.CKT) file and the test program (.TPX) [1] file. Since the actual test nails (also referred to as target nails) are not known at this time, the user inserts temporary nail numbers in the ADAPTOR section (i.e., %ADAPTOR:227x TEMP) of the .CKT file. CKTGEN can be used to prepare the circuit description and ATG is used to compile the .CKT file and develop the test program.

---

[1] An X is used in certain file extensions (.TPX, .ATX, .IDX) to indicate that the file contains temporary nail numbers. When nail assignment replaces the temporary nail numbers with target nail numbers, it automatically renames the files .TPG, .ATO, .IDD.

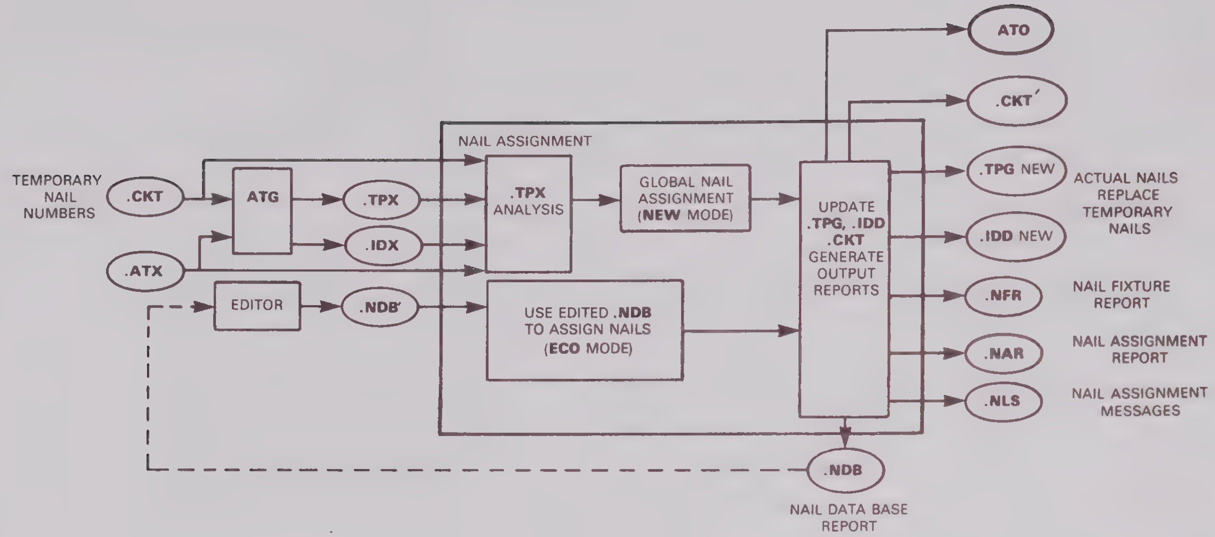

ATO
.CKT′
NAIL ASSIGNMENT
TEMPORARY NAIL NUMBERS
.CKT
ATG
.TPX
.TPX ANALYSIS
GLOBAL NAIL ASSIGNMENT (NEW MODE)
UPDATE .TPG, .IDD .CKT GENERATE OUTPUT REPORTS
.TPG NEW
ACTUAL NAILS REPLACE TEMPORARY NAILS
.IDX
.IDD NEW
.ATX
EDITOR
.NDB′
USE EDITED .NDB TO ASSIGN NAILS (ECO MODE)
.NFR
NAIL FIXTURE REPORT
.NAR
NAIL ASSIGNMENT REPORT
.NLS
NAIL ASSIGNMENT MESSAGES
.NDB
NAIL DATA BASE REPORT

Figure 8-1. Test Set Preparation/Modification.

Before running ATG, the user can also create an optional .ATX file (using temporary nail numbers) to specify test options for ATG to make to the test program while it is being developed.

The circuit description is then compiled via the ATG "Prepare" options. From the compiled circuit description and the user- generated ATX file, ATG then develops the test program (.TPX) file and circuit data base (.IDX) file.

Once an acceptable test program is available, the user specifies the .TPX file as an input to the nail assignment software. The user can also specify other files (.CKT, .IDX, .ATX), as desired. Although not essential to the operation of nail assignment, it is recommended that the .CKT and .IDX files be updated, for user convenience. An optional .NDB file can also be used to designate special nail assignments that should be made. This is especially useful when an existing test set is being modified. The .NDB file can be developed either manually by the user or automatically by CKTGEN when the circuit description is changed.

Using these input files and the internal nail-assigning algorithm, the software assigns new target nail numbers or checks any previous assignments made by the user. Once the nail assignment function is complete, the system updates the input files (unless specifically disabled by user option) and creates various output reports that can be used to wire the fixture (.NFR), to keep a record of nail assignments (.NAR), to serve as a data base from which to change subsequent nail assignments (.NDB), or to evaluate the nail assignment process (.NLS). These reports are described more fully at the end of this section.

When the nail assignment process is complete, the test program source code (.TPG file) is translated into object code (via Translate mode), at which time the translator makes further checks for nail conflicts. After the test fixture has been wired, board testing can take place under control of the Run-Time system. Since the nail assigner and translator can only check for test program nail conflicts at compile-time, it is possible for the Run-Time system to discover conflicts added later. This would normally only occur if user modifications had been made to an ATG-developed test program.

NOTE

Before using any of the monitor modes to develop or change a test set, make sure that the "Target System" option on the monitor SETUP page specifies the desired system. The target system is the system for which the test set is being prepared.

The nail assignment software not only assigns nails to new test sets, but also performs the other related functions shown in the following table. More detailed descriptions of these nail assignment functions appear in the referenced paragraphs.

Nail Assignment Function	Nail Assignment Monitor Mode/Options	Ref Para
Preparing a new test set	NEW mode	8.3
Modifying an existing test set	ECO mode	8.4
Using an existing 2270 or 2275 test set on another test system	2270-EMULATION mode or 2275-EMULATION mode	8.5
Developing a new test set common to both 2270 and 2271	"Pure-pin range" option	8.6

8.2 NAIL TYPES

8.2.1 General

The various nail types associated with nail assignment are listed in Table 8-1 and briefly described in the subsequent paragraphs. The nail symbols (H, F, CD, etc) shown in the table are nail type designators recognized by the nail assignment software and used by the software in its output nail assignment reports.

Table 8-1
NAIL TYPES

Nail Type	Symbol	2271		2272	
		n	MUX Ratio	n	MUX Ratio
Hybrid	Hn	1-960	(4-to-16)	1-3584 (see Note)	(2-to-16)
Fast	Fn	4001-4320	(2-to-8)	4001-4864 (see Note)	(2-to-8)
Clock Drive	CDn	1-7, 9-15	(1-to-4)	1-16	(1-to-4)
Clock Sync	CSn	1-8	(1-to-8)	1-8	(1-to-8)
Trigger	Tn	1-22,25,26	(1-to-4)	1-28	(1-to-4)

Note: The maximum number of hybrid and fast nails specified here are not both available in the same 2272 system at the same time. For example, each group of 32 fast nails in a 2272 system reduces the maximum number of hybrid nails by 128.

The fast, clock drive, clock sync and trigger nails are all part of the SpeedPlus option. Of these, the clock and trigger nails do not support normal testing (i.e., they are not connected to driver/sensors). Therefore, an additional hybrid or fast test nail may have to be connected to a clock/trigger node for test purposes. To avoid permanently connecting both types of nails to the same node (at the same time) and possibly degrading the performance of that node due to extra loading, the test nail can be wired to the node via an external relay (located on the Clock Generator and Trigger boards within the system).

The diagram in Figure 8-2 shows a typical test nail-to-clock/trigger nail connection. Each clock/trigger nail has 2 connections on the receiver block: one direct to the clock or trigger circuit in the system, the other through an external disconnect relay on the board. The test nail gets wired to the relay rather than directly to the UUT. Special wiring instructions for this arrangement appear in the nail assignment output reports, and are described in detail in the Test Fixture Manual. A DECLARE NAILBIND statement (described in the Test Programming Manual) is used in the test program to inform the run-time system of the connection.

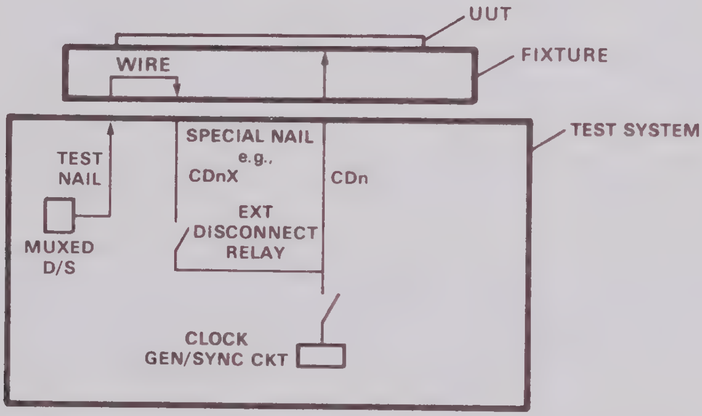


Figure 8-2. Test Nail-to-Clock/Trigger Nail Connection.

During testing, the system closes the external relay whenever the node is being tested. At all other times, the relay is kept open, thereby disconnecting the test pin from the node.

8.2.2 Analog-only/Digital-only Nails

Analog-only nails connect only to the analog instrument bus; digital-only nails connect only to the digital driver/sensors. These "pure-pin" nails are not multiplexed and are found only in 2270 systems. The only time they need be specified in the nail assignment process is when you wish to develop a new test set compatible with both a 2270 and 2271 system (para 8.6).

8.2.3 Hybrid Nails

Hybrid nails are the standard test nails used in all of GenRad's multiplexed test systems (and are also available in the 2270 system). Each hybrid nail can be connected either to the analog instrument bus or to a driver sensor. In the 2271 and 2275 systems, groups of 16 hybrid nails share 4 driver/sensors (i.e., 4-to-16 multiplexing). In the 2272 system, groups of 16 nails share 2 driver/sensors (2-to-16 multiplexing).

8.2.4 Fast Nails (SpeedPlus Option Only)

Standard hybrid nails can perform digital tests at rates up to 280,000 test steps per second. The SpeedPlus option provides special (fast) hybrid nails that can execute up to 5 million test steps per second. Groups of 8 fast nails share 2 high-speed driver/sensors, for a 2-to-8 multiplexing ratio.

8.2.5 Clock Drive Nails (SpeedPlus Option Only)

These special nails can be used to supply programmable clock inputs to the UUT. A clock generator circuit in the test system generates up to 4 separate clock signals and multiplexes them to 14 (2271) or 16 (2272) clock driver nails, for a 1-to-4 multiplexing ratio. Clock frequencies are programmable from 20 KHz to 20 MHz. Drive and sense strobes can be programmed to occur at any time increment during the clock cycle.

8.2.6 Clock Sync Nails (SpeedPlus Option Only)

The clock sync nails are for handling test procedures where each digital test step must be synchronized with a clock on the UUT (e.g., when a UUT clock cannot be disabled). Any one of 8 clock sync nails (in a 1-to-8 multiplexing arrangement) can be used for this purpose. The selected sync signal controls the internal timing of the test system during the burst.

8.2.7 Trigger Nails (SpeedPlus Option Only)

The trigger nails can be used to synchronize a test with a particular bus cycle or other unique event occurring on the UUT. Event triggering allows the test system to monitor control pins on the UUT and align its test strobes with the occurrence of specific conditions. Up to 7 separate conditions can be monitored. The trigger is generated when all of the specified conditions are satisfied.

Trigger nails are multiplexed 1-to-4 (1 trigger sensor to 4 trigger nails). A 2272 with SpeedPlus option has 28 trigger nails; a 2271 has 24.

8.3 PREPARING A NEW TEST SET

The overall procedure for developing the test program and preparing the test fixture is covered in Section 4. The major steps in using Nail Assignment mode for preparing a new test set are described in this paragraph.

8.3.1 Run Nail Assignment

Once a satisfactory test program is obtained, that is, one that has been developed from an accurate circuit description and references all devices to be tested, call up the Nail Assignment monitor page. (A description of each nail assignment option can be found in para 8.7). Prepare a new test set by selecting the options shown in Example.

Example

Mode:	NEW
Lowest hybrid nail = Highest hybrid nail =	Range of nails available in the particular system being used.
Update files [YES,NO]	YES
High-Speed Enhancement [NO,YES]	YES if you are using SpeedPlus option; NO, otherwise
Pure pin range [YES,NO]	NO
NFR type[FINAL,TEMP]	Select format for wire-wrap report (See Figure 8-8)
Low-fast nail = High-fast nail =	For SpeedPlus option only, specify the range of fast nails available for nail assignment
TPXin =	filespec for ATG-developed test program
IDX [Y,N] IDXin =	filespec for ATG-provided data file
CKT [Y,N] CKTin =	filespec for user-created circuit description file
ATX [N,Y] ATXin =	filespec for user-created options file
NDB [N,Y]	N
TPGout =	filespec for modified test program
IDDout =	filespec for modified data file
CKTout =	filespec for appended circuit description file
ATOout =	filespec for modified options file
NDBout =	filespec for nail data base output
Report =	device for output reports
Temp =	device for temporary files

Running the nail assignment software with these options will cause the system to perform the following functions:

1. Analyze the test program and assign nails as appropriate to avoid conflicts; a global assignment is made to all test nails.

2. Replace the temporary nail numbers in .TPX, .IDX, and .ATX with actual fixture nail numbers, and rename those files .TPG, .IDD, .ATO. (See Figure 8-3 for examples of a .TPX and a .TPG file.)

3. Add an ADAPTOR:227x section to the .CKT file (Figure 8-4), specifying the target nail numbers rather than the temporary numbers. (Note: If an ADAPTOR:227x section already exists in the .CKT file, it will be deleted.

4. Output the 4 nail assignment reports (.NFR, .NAR, .NDB, .NLS) described and illustrated at the end of this section.

.TPX (Before Nail Assignment)

```
R9:       MEAS R AT(21:13:22:22) DLY=167.042U RDLY=40.878U
            CMSG='R9' DMSG='330' HI=353.337 LO=312.329
              MAX=332.833;

Z1:       MEAS R AT(21:1:22:22) DLY=171.003U
            CMSG='Z1' DMSG='3.3K' HI=3.503K LO=3.105K
              MAX=3.304K;

Z1_1:     MEAS R AT(21:4:22:22) DLY=171.003U
            CMSG='Z1' DMSG='3.3K' HI=3.503K LO=3.105K
              MAX=3.304K;
```

```
/** TEST PROGRAM FOR IC U6 (7400)
PIN NAIL COMMENT
  4 60
  5 59
  6 61
 **/
        IC(59,60) IL(59) IH(60) OS(61) OH(61);
        IH(59) OL(61);
        IL(60) OH(61);
        IL(59);
        ID(59,60) OI(61);
```

TEMPORARY NAILS

.TPG (After Nail Assignment)

```
R9:       MEAS R AT(81:72:97:97) DLY=167.042U RDLY=40.878U
            CMSG='R9' DMSG='330' HI=353.337 LO=312.329
              MAX=332.833;

Z1:       MEAS R AT(81:17:97:97) DLY=171.003U
            CMSG='Z1' DMSG='3.3K' HI=3.503K LO=3.105K
              MAX=3.304K;

Z1_1:     MEAS R AT(81:49:97:97) DLY=171.003U
            CMSG='Z1' DMSG='3.3K' HI=3.503K LO=3.105K
              MAX=3.304K;
```

```
/** TEST PROGRAM FOR IC U6 [7400]
PIN NAIL COMMENT
  4 60
  5 59
  6 61
 **/
        IC(52,193) IL(52) IH(193) OS(194) OH(194);
        IH(52) OL(194);
        IL(193) OH(194);
        IL(52);
        ID(52,193) OI(194);
```

ASSIGNED NAILS

Figure 8-3. Sample Test Program (Partial) Before and After Nail Assignment.

```
/* 2720all      For: 2271       Generated by:   NASMN436
   Tue 02-Aug-83 06:04:12                Mode:   Eco Update      */

%ADAPTER:2271;

CD1        = N111;
CD5        = N201;
T2         = N254;
T6         = N117;
T1         = N252;
T7         = N252;
T5         = N220;
T3         = N220;
T17        = N219;
T21        = N221;
T13        = N235;
T14        = N807;
T17        = N232;
T21        = N233;
T25        = N234;
H1         = N632;
H1         = N111;
H2         = N476;
H2         = N390;
H3         = N570;
H4         = N477;
H5         = N878;
H6         = N772;
H7         = N596;
H8         = N282;
H9         = N120;
H10        = N235;
H11        = N150;
H12        = N466;
H13        = N767;
H14        = N720;
H15        = N468;
H16        = N91
H17        = N574;
H18        = N309;
H19        = N519;
H20        = N768;
H21        = N671;
H22        = N679;
H23        = N877;
H24        = N770;
```

NOTE

During the nail assignment process, this new %ADAPTOR section showing assigned nails is added to the .CKT file. Refer to Section 6.

Figure 8-4. Revised Circuit Description (.CKT) from Nail Assignment.

In systems that have the SpeedPlus option, several different types of nails are available for assignment (refer to para 8.2). In general, the nail types specified or required by the input files (.TPX, .CKT, .IDX) are not changed by the nail assignment process. If .TPX is the only input file, nail assignment will assign nail types based on the minimum requirements needed to run the test program effectively.

If both the .TPX and .IDX files are specified as inputs, nail assignment has 2 sources of nail-type information and may have to make a decision if the two sources conflict. In NEW mode, if a conflict exists, the following action will take place:

1. If the .CKT (or .IDX) file defines a nail as being a particular type (hybrid, fast, etc), but the nail is never used that way in the test program, nail assignment will leave the nail type unchanged (i.e., the way it was defined in .IDX).

2. If a nail is used as a fast nail in .TPX but is defined as a hybrid nail in .CKT (or .IDX), nail assignment will ignore the .CKT definition and make that nail a fast nail.

Other special nail assignments may also be made under the following conditions:

1. If the system has too few hybrid nails for a particular test, nail assignment will replace an input hybrid nail with a fast nail.

2. If an input .NDB file is provided, nail assignments may be made in accordance with the special instructions in that file.

8.3.2 Update .CIR File in CKTGEN

If CKTGEN was used to develop the circuit description, it is recommended that the CKTGEN data base file (.CIR) be updated after nail assignment is complete. Doing this will allow you to use CKTGEN again in making any future changes to the test set.

To update .CIR, call up the CKTGEN monitor page, select the READ mode option, specify the updated .CKT file as an input and the new .CIR file to be created as an output, then run the task. A new .CIR file will be created, reflecting the newly assigned nail numbers. Refer to Section 6 for details on CKTGEN.

8.3.3 Prepare and Wire the Test Fixture

Once the Nail Fixture Report (.NFR file) is available from the nail assignment software, the information in that report can be used to wire the fixture. Refer to the appropriate GenRad Test Fixture Kit Assembly Manual for specific instructions in preparing and wiring the test fixture.

8.4 MODIFYING A TEST SET

Once a new test program has been developed, test nails have been assigned, and the test fixture has been wired, a known-good board can be used to debug the new test set. During this debugging phase, or later during production testing, the user may find it necessary to assign new nails to the test set, e.g., in the case of wiring errors in the fixture or engineering changes to the pc board being tested. To help the user perform this function, the nail assignment ECO mode checks the modified test program for nail conflicts, resolves conflicts, updates the test set files (if not specifically disabled by the user), and generates updated reports to record the new nail assignments.

Figure 8-5 illustrates the typical steps involved in modifying test program nail assignments. Basically, the procedure consists of updating the input files (.TPG, .CKT, .IDD,.ATO) to reflect the changes, creating an input .NDB file showing what changes have been (or should be) made, then running nail assignment ECO mode. Each step in the process is further described in the following paragraphs.

NOTE

When modifying a test set, it is good practice not to delete existing nails, but to implement the change by adding new (unused) nails. The nail assigner never deletes existing nails in ECO mode.

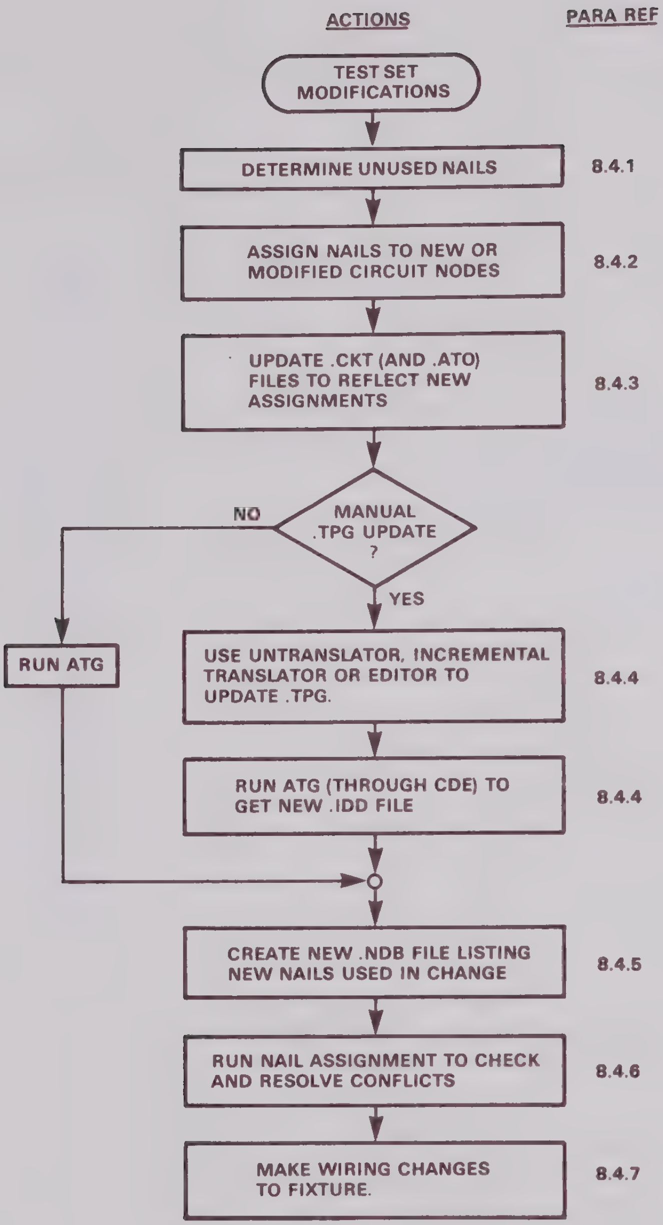

ACTIONS
PARA REF
TEST SET MODIFICATIONS
DETERMINE UNUSED NAILS
8.4.1
ASSIGN NAILS TO NEW OR MODIFIED CIRCUIT NODES
8.4.2
UPDATE .CKT (AND .ATO) FILES TO REFLECT NEW ASSIGNMENTS
8.4.3
MANUAL .TPG UPDATE ?
NO
YES
RUN ATG
USE UNTRANSLATOR, INCREMENTAL TRANSLATOR OR EDITOR TO UPDATE .TPG.
8.4.4
RUN ATG (THROUGH CDE) TO GET NEW .IDD FILE
8.4.4
CREATE NEW .NDB FILE LISTING NEW NAILS USED IN CHANGE
8.4.5
RUN NAIL ASSIGNMENT TO CHECK AND RESOLVE CONFLICTS
8.4.6
MAKE WIRING CHANGES TO FIXTURE.
8.4.7

Figure 8-5. Modifying a Test Set.

If a nail assignment output file is lost, ECO mode can also be used to recreate that file. Simply run ECO mode, using the original input files (.TPX, .IDX, .CKT) and the .NDB output file from the first run (modified, if necessary, to match actual fixture wiring).

8.4.1 Determine Unused Nails

Prior to making nail assignment changes to the test set, the user must determine what test nails are available for the new assignments. This information is indirectly available in the nail assignment reports (.NFR, .NAR, .NDB), which list all nails that have been assigned. A list of available nails can also be obtained via a FREE command in Debug sub-mode. From the lower-level monitor, typing FREE and pressing the RETURN key causes the system to display a list of available nails. Refer to Section 9 for a description of the FREE nail command.

8.4.2 Assign Available Nails to New Nodes

On a circuit schematic, specify a new nail (from the list of unused nails) for each new node. Conflicts not anticipated at this stage, will be detected later by the nail assigner or translator.

If nail conflicts are detected and nail assignments changed, be sure to record those changes on the schematic.

8.4.3 Update .CKT (and .ATO) Files

Using information from the marked-up schematic diagram, update the circuit description (.CKT) file (and the .ATO file, if one is being used) to reflect the circuit changes. When changing the ADAPTOR section, add the new nail connections to the %ADAPTOR:227x section. The other (temporary) ADAPTOR section is ignored by the nail assigner during ECO mode.

If the .CIR file (used by CKTGEN) has been updated to reflect the latest nail assignments (refer to para 8.3.2), CKTGEN can be used to make the changes to the circuit description. CKTGEN also automatically develops an input .NDB file for nail assignment, showing what temporary nails CKTGEN has added to the target ADAPTOR section of the .CKT file. This input allows nail assignment to distinguish between new temporary nails introduced by this change and target nails that already existed in the circuit description. Refer to Section 6 for more details on CKTGEN.

8.4.4 Update .TPG (and .IDD) Files

Depending on how extensive the change, the .TPG file may be modified either manually by the user or automatically by ATG.

For large changes that cannot be incorporated manually, run ATG with option "Nail State=TARGET." The initial stage of ATG will compile the .CKT file and produce a new .IDD file based on the updated circuit description. The completion of ATG will produce a new .TPG file that will serve as an input to Nail Assignment.

For small changes, the user can modify the .TPG file directly via Diagnose (Debug) mode or Edit mode. If the .CKT file requires changes that cannot be specified in the .NDB file, return to the ATG monitor page and rerun the "Circuit prepare" and "IDD prepare" options with "Nail State=TARGET."

8.4.5 Create New .NDB File (e.g., ECO.NDB)

To inform the nail assigner of what new nails were added to the test program, the user must create a .NDB (Nail Data Base) input file, listing each newly assigned nail. The recommended format for this file is shown in the following example.

```
/*ECO.NDB FOR BOARD ECN#1*/
%NAILMAP 2271 ECO;
801 -> X;
817 -> X;
834 -> 900;  /*ALREADY ADDED TO FIXTURE; NO NAIL CONFLICTS*/
```

In this example, 801, 817, and 834 are new nails. The nail assigner interprets the entry

```
801 -> X;
```

as follows: If new nail 801 does not conflict with the other nails, let the assignment stand. If there is a conflict, replace the nail with another unused nail that does not cause a conflict, and report the change.

The entry

```
834 -> 900;
```

indicates that a fixture wiring change was made. Nail assignment interprets this entry by replacing all occurrences of nail 834 with nail 900.

If the SpeedPlus option is used, the following entry format allows the user to force conversions between hybrid and fast nails.

```
H51 -> FX
```

This forces the software to make nail 51 a fast-type nail even though the input .TPG may not require it. This type of conversion is allowed only between hybrid and fast test nails and not between a test nail and any of the special clock/trigger nails.

8.4.6 Run Nail Assignment

Call up the Nail Assignment monitor page and specify the options shown in Example.

Example

Mode:	ECO
Lowest hybrid nail =	nail range of the particular test system being used
Highest hybrid nail =	
Update files [YES,NO]	YES
High-Speed Enhancement [NO,YES]	YES if you are using the SpeedPlus option; otherwise, NO
Pure-pin range [YES,NO]	NO
NFR type	Select format for wire-wrap report (See Figure 8-9)
Low-fast nail =	For SpeedPlus option only, specify the range of fast nails available for nail assignment
High-fast nail =	
TPGin	filespec of debug-modified test program
IDD [Y,N] IDDin =	filespecs of input files to be updated
CKT [Y,N] CKTin =	
ATO [N,Y] ATOin =	
NDB [N,Y] NDBin =	

TPGout = filespec for updated test program

IDDout = filespecs for updated output files
CKTout =

ATOout =
NDBout =
Report = device on which to output reports
Temp = device on which to store temporary files

Running nail assignment with these options will cause the system to:

1. Check for conflicts in the newly-assigned nails. If necessary, nail assignment will add new nails to a node, but will not delete existing nails.

2. Resolve the conflicts (if possible) and update the .TPG, .CKT, .IDD, and .ATO files

3. Create new nail reports (.NFR, .NAR, .NDB, .NLS).

8.4.7 Make Wiring Changes to Fixture

The new nail assignments will appear in all the reports. Use the .NFR file as the new wiring list and make the indicated wiring additions to the fixture.

NOTE

> If the system changes a user-assigned nail to resolve a conflict, be sure to record that change on the circuit schematic. Changes are flagged in both the .NFR and .NAR reports, by asterisks (**).

8.5 USING 2270 or 2275 TEST SET ON ANOTHER SYSTEM

Nail Assignment mode can also help the user who wants to use an existing 2270 or 2275 test set on another type of system. Optional fixture converters are available for this purpose. These converters are specially designed to minimize the chance of nail assignment conflicts occurring when interfacing one type of fixture to another type of system.

Nail assignment EMULATION mode is used to check the existing test program for possible nail conflicts. Via this mode, 2270 test programs can be adapted to run on a 2271 or 2272 system, and 2275 programs can be adapted to run on a 2272 system.

In EMULATION mode, the nail assigner examines test nail assignments in the existing test program, in conjunction with the fixed wiring of the converter. It then reports on any multiplexing conflicts within the test program, suggests wiring additions to the existing fixture to resolve conflicts, and updates the input files at the option of the user.

To use the EMULATION mode, call up the Nail Assignment monitor page and specify the options shown in Example.

Example

Mode	2270-EMULATION or 2275-EMULATION
Lowest hybrid nail =	range of 2271 (or 2272) test nails to be used in the emulation
Highest hybrid nail =	
Update files	YES
NFR type [FINAL, TEMP]	select format for wire-wrap report (see Figure 8-8)
TPGin=	filespec for 2270 or 2275 test program
IDD [Y,N] IDDin=	filespec for 2270 or 2275 diagnostic data file
NDB [N,Y] NDBin=	Indicates if a .NDB file is being used. Typically, N the first time emulation is run, and Y whenever it is re-run for the same test set. (See following paragraph).
TPGout=	filespec for 2271 (or 2272) test program
IDDout=	filespec for 2271 (or 2272) diagnostic data file
NDBout=	filespec for output nail data base
Report=	device for output reports (.NFR, .NAR, .NLS)
Temp=	device for temporary files

The first time a test program is run through EMULATION mode, specify the test program input (TPGin) and the IDD input (IDDin). When nail assignment is complete, it will list any new wiring additions that must be made to the 2270 or 2275 fixture in the .NDB (and .NFR) reports. Keep this .NDB file for future reference. The .CKT file does not get updated.

If subsequent changes are made to the test set, manually insert these changes in the .CKT file and run ATG to develop an updated test program (and .IDD file). Then, re-run EMULATION mode, specifying the updated .TPG, and .IDD files as inputs, along with the .NDB file saved from the previous emulation run.

The resulting .NFR file will specify only the new fixture wires required for this recent change. The resulting .NDB file, however, will contain all the new fixture wires specified by both passes through nail assignment. In other words, the list of new wires recorded in .NDB is cumulative for each pass through nail assignment. All the new wires are indicated by ** in the .NAR. The latest version of the .TPG file, after emulation is run, will be the 2271 or 2272 version. The 2270 or 2275 version must be accounted for so that it is not lost for use on the 2270 or 2275 system.

NOTE

When making changes to a 2270 or 2275 test set for use on a 2271 or 2272 system, always use EMULATION mode (not ECO mode) to check the nail assignment, and always re-translate the .TPG file to obtain a new object code test program.

8.6 DEVELOPING A COMPATIBLE 2270-2271 TEST SET

If a new test set is being developed and requires that the same fixture be used on both a 2270 and 2271, use the "Pure pin range" option on the Nail Assignment monitor page. The "Pure pin range" option lets the system assign nails to a 2270 test program in such a way that the resulting test set can be used on both a 2270 and 2271 system.

To use this feature, first develop the test program (.TPG) and associated files (.CKT, .IDD, and .ATO) for the 2270 test set, specifying temporary nail numbers. Call up the Nail Assignment monitor page and specify the options shown in Example.

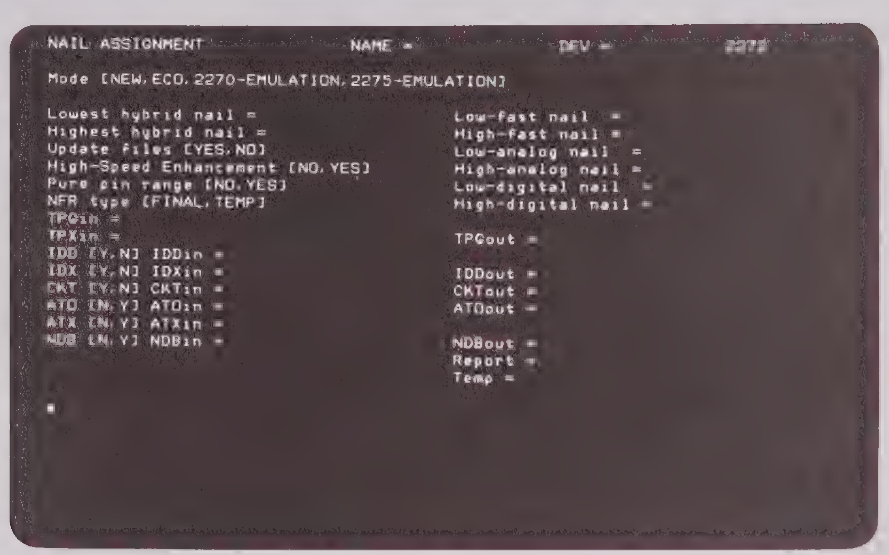
Example

Mode	NEW
Lowest hybrid nail =	range of hybrid nails in the
Highest hybrid nail =	2270 test system
Update files [YES,NO]	YES
High-Speed Enhancement [NO,YES]	NO
Pure pin range [NO,YES]	YES
NFR type	Select format for wire-wrap report (See Figure 8-8)
Low analog nail =	range of pure analog nails in the
High analog nail =	2270 test system
Low digital nail =	range of pure digital nails in the
High digital nail =	2270 test system
TPXin=	filespec for the 2270 test program
IDX[Y,N] IDXin=	
CKT[Y,N] CKTin=	filespecs for associated 2270 input files
ATX[N,Y] ATXin=	
NDB[N,Y]	NO
TPGout=	
IDDout=	filespecs for updated files
CKTout=	
ATOout=	
NDBout=	filespec for output nail data base
Report =	device for output reports
Temp =	device for temporary files

When the nail assignment function is complete, use the Nail Fixture Report (.NFR file) to wire the test fixture. The system-specified nail assignments will allow the resulting test set to be used either on the 2270 system for which it was developed or on a 2271 system.

8.7 NAIL ASSIGNMENT MONITOR PAGE OPTIONS

Following is a description of the various options in the Nail Assignment monitor page (Figure 8-6).



```
NAIL ASSIGNMENT              NAME =               DEV =          2272

Mode [NEW,ECO,2270-EMULATION,2275-EMULATION]

Lowest hybrid nail =                          Low-fast nail    =
Highest hybrid nail =                         High-fast nail   =
Update files [YES,NO]                         Low-analog nail   =
High-Speed Enhancement [NO,YES]               High-analog nail  =
Pure pin range [NO,YES]                       Low-digital nail   =
NFR type [FINAL,TEMP]                         High-digital nail  =
TPGin =
TPXin =                                       TPGout =
IDD [Y,N] IDDin =
IDX [Y,N] IDXin =                             IDDout =
CKT [Y,N] CKTin =                             CKTout =
ATO [N,Y] ATOin =                             ATOout =
ATX [N,Y] ATXin =
NDB [N,Y] NDBin =                             NDBout =
                                              Report =
                                              Temp =

■
```

Figure 8-6. Nail Assignment Monitor Page.

Mode [NEW,ECO,2270-EMULATION,2275-EMULATION]

[NEW] The default condition, causes the system to assign actual nail numbers to all the temporary nails in the .TPX input file in accordance with an internal nail assignment algorithm. This mode is typically used when preparing a new test set, i.e., one in which no previous nail assignments have been made.

[ECO] This mode helps the user make nail assignment changes to the test set. After the user makes the desired changes to the .TPG, .CKT, .IDD, and .ATO files, running ECO mode will do the following:

1. Check for MUX conflicts in the newly-assigned nails.

2. If specified by the "Update files" option, resolve the conflicts and update the input files (.TPG, .CKT, .IDD, .ATO). Preferred nails for resolving conflicts can be specified in the .NDBin file.

3. Generate new nail reports (.NFR, .NAR, .NDB, .NLS)

[2270-EMULATION] or [2275-EMULATION]

This mode is aimed at making a 2270 or 2275 test program run on another type of test system equipped with a special fixture converter. Making use of the converter wiring information, the nail assigner checks the test program for nail conflicts, and resolves them, if possible.

Lowest hybrid nail=
Highest hybrid nail=

Used to specify a range of test fixture nails available to the nail assignment software. In NEW mode, this range would typically include all the nails in the system (minus a group of nails reserved for spares, if desired). In ECO mode, the specified range should include nails not yet used in TPG. If the "Pure pin range" option is used, then the hybrid nail range refers to the range of hybrid nails in the 2270 system associated with that "Pure pin range" option. In EMULATION mode, this range is all the hybrid nails available on the target system.

NOTE

If left unspecified (and "Pure pin range" is [NO]), the hybrid nail range will default to the maximum nail range available in the system, regardless of whether the system being used actually contains that many nails.

Update files [YES,NO]

[YES] The default condition, causes the system to update the input .TPX, .IDX, .CKT, and .ATX files after the nail assignment function is complete. Updated files are stored in the filespecs specified in the TPGout =, IDDout =, CKTout =, and ATOout = options.

[NO] Prevents these files from being automatically updated.

High-Speed Enhancement [NO,YES]

Selects the SpeedPlus option available on the 2271 and 2272 systems. In addition to hybrid test nails, this option makes available fast test nails, clock drive nails, clock sync nails and trigger nails.

Pure pin range [NO,YES]

This option causes the nail assignment software to try to assign nails to a 2270 test program, so that the resulting test set can be used on both a 2270 and 2271 system. The option would not be used when preparing or modifying a 2271, 2272, or 2275 test set. This option only appears on the display when the target system is set to 2271 and the mode is NEW or ECO.

When using this option, be sure to specify the range of pure analog, pure digital and hybrid nails used in the 2270 test system.

NFR type [FINAL,TEMP]

[FINAL] The resulting fixture report emphasizes nails actually assigned in the target system's program file (see Figure 8-8, page 1).

[TEMP] The resulting fixture report emphasizes the nails temporarily assigned in the test program source file (see Figure 8-8, page 2).

Low-fast nail=
High-fast nail=

Used only with the "High-Speed Enhancement" option to specify the range of fast nails available to the system. Fast nail numbers must be in the range of 4001 to 4320 (2271) or 4001 to 4864 (2272). Since fast nails are available in groups of 8, the low-fast nail must be either 4001, 4009, 4017, etc, and the high-fast nail must be some multiple of 8 above the starting number (e.g., 4016, 4024, 4032, etc). A blank input causes that nail specification to default to 0. Blanks (or 0's) are allowed to indicate that no fast nails should be assigned.

Low analog nail=
High analog nail=

Used only with the "Pure pin range" option to specify the range of pure analog test nails in the 2270 test system.

Low digital nail=
High digital nail=
Used only with the "Pure pin range" option to specify the range of pure digital test nails in the 2270 test system.

TPGin= (or TPXin)
Required input to the Nail Assignment page. Specifies the test program input file to the nail assigner. When used in NEW mode, this file (called .TPX) contains all temporary test nails which the nail assigner replaces with actual nails (if enabled by the "Update files" option). When used in ECO mode, the TPGin file is assumed to contain actual nail numbers, which the nail assigner checks for conflicts and tries to resolve, if possible. When used in EMULATION mode or with the "Pure pin range" option, TPGin must specify the 2270 test program file. (If NEW mode is selected, TPGin appears on the monitor page as TPX, indicating that the file contains temporary nail numbers.)

IDD[Y,N] IDDin= (or IDX[Y,N] IDXin=)
CKT[Y,N] CKTin=
ATO[N,Y] ATOin= (or ATX[N,Y] ATXin=)
[Y] These options allow the user to specify each of 3 other input files associated with TPGin. All 3 files are updated by the nail assigner if enabled by the "Update file" option.

1. The .IDD (In-circuit Diagnostic Data) file, generated by the Circuit Description Extractor (CDE) phase of ATG, contains the names and connections of all components on the board. In ECO mode, the nail assigner assumes, as unused nails, all nails not in the specified .IDD or .TPG file. The user can indicate preference of what unused nails to assign via the "Lowest/Highest hybrid nail" option and by inputs to the NDBin file. (If NEW mode is selected, IDDin appears on the monitor page as IDX, indicating that the file contains temporary nail numbers.)

2. When used in NEW mode, the .CKT (Circuit Description) file is the user-developed circuit description input to ATG, containing all temporary nail numbers. (Note that ATG uses this .CKT file to develop the .IDD file). If enabled by the "Update files" option, the nail assigner, upon completion of its nail assignment function, adds an additional ADAPTOR section to the .CKT file, containing actual test nail numbers, as follows:

   ```
   %ADAPTOR:227x;
    (System-developed ADAPTOR section with
    actual nail numbers)
   ```

   Note: If the .CKT file already has a %ADAPTOR:227x section (specified for a particular system), then the nail assigner deletes that section and replaces it with a new %ADAPTOR:227x section.

   In ECO mode, the nail assigner updates only the %ADAPTOR:227x section (if enabled) to reflect nail re-assignments.

3. The .ATO (Automatic Test Options) file is an optional input file whereby ATG makes user-specified changes to the test program as it is being created. As with the .IDD file, in NEW mode, the .ATO file contains temporary nail numbers which the nail assigner can update to

actual nail numbers. In ECO mode, the .ATO file is assumed to contain actual nail numbers. (If NEW mode is selected, ATOin appears on the monitor page as ATX, indicating that the file contains temporary nail numbers.)

[N] N indicates no input file.

NDB [N,Y] NDBin =

In ECO mode (and to some extent in NEW mode), the user can exert some influence on the internal nail assignment algorithm via the NDBin (Nail Data Base) file. In ECO mode, the NDBin file is used primarily to inform the nail assigner of what new nails were added to the test program, but can also be used to force certain assignments. The recommended format for doing this is as follows:

```
%NAILMAP 227x ECO;
801 -> X;
```

meaning that nail 801 is new and can be replaced by any other unused nail if a conflict exists. Other formats for entering data in the .NDB file include:

```
376 -> 833;
833 -> 376;
```

which causes the nail assigner (in ECO mode) to interchange nails 376 and 833 in the test program (and associated files), and

```
118 -> 118 201;
```

which causes a new nail (201) to be added to the same circuit node as the existing nail (118), and

```
H87 -> FX
```

causes nail assignment to make nail 87 a fast nail (SpeedPlus option only) even though the test program does not require it to be a fast nail.

On a limited basis, NDB can also be used in NEW mode, for example, to force particular nail assignments to the PROBE and SYNC (TEST STEP N) connectors or to specify the nail offset in a split fixture. Specifying the following .NDB file during NEW mode:

```
%NAILMAP 227x NEW;
PROBE ->  1;
SYNC  -> 17;
```

causes the nail assigner to assign nail 1 to the PROBE, nail 17 to the SYNC pulse connector. Note, however, that placing too many restrictions on the nail assigner via .NDB during NEW mode could possibly prevent the global nail assignment function from being performed. Nails forced by the user in the .NDB are not checked by nail assignment for conflicts. Conflicts would appear as a fatal error in the translator.

In EMULATION mode, the .NDB file can be used to inform nail assignment of previous changes to existing 2270 or 2275 fixture wiring. Simply use the .NDB output file from the previous nail assignment run in EMULATION mode. The file header in this case would be

%NAILMAP 2270 (or 2275) EMULATION;

TPGout=
IDDout=
CKTout=
ATOout=
NDBout=

These options specify the file names for the updated files. An .NDBout file is always created. The other files are produced only if enabled by the "Update files" option.

Report =

Specifies the device on which to store the nail assignment reports (.NFR, .NAR and .NLS). These reports are described in para 8.8. The Report device defaults to the "DEV =" device.

Temp=

Specifies the device on which to store the temporary files created by Nail Assignment. Defaults to the "DEV =" device.

8.8 NAIL ASSIGNMENT AND FIXTURE REPORTS

8.8.1 General

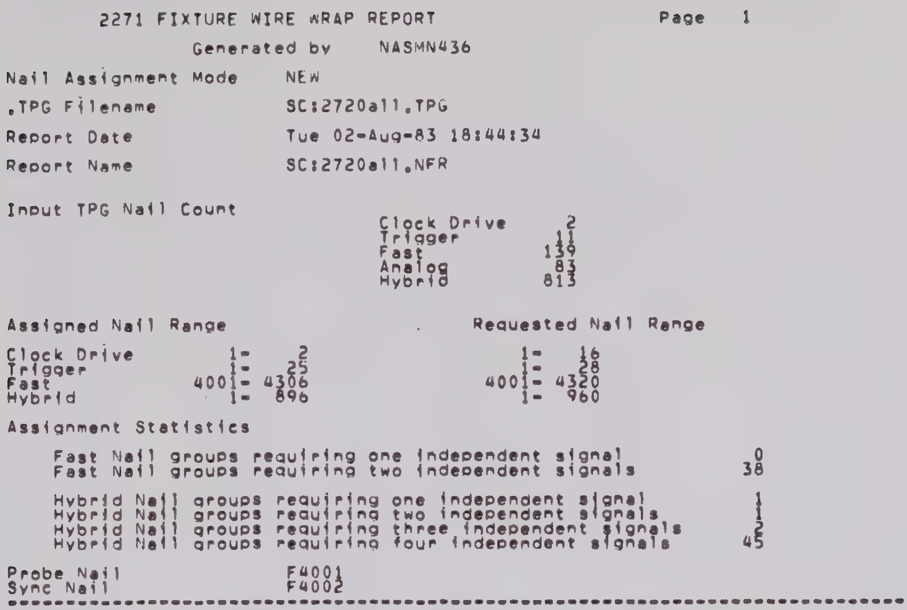
The nail assignment process produces 4 output report files: Nail Fixture Report (.NFR), Nail Assignment Report (.NAR), Nail Data Base report (.NDB) and a Nail Assignment Listing (.NLS). A common header page showing input and assigned nail ranges and assignment statistics is included at the beginning of each report, except .NLS (see Figure 8-7). In addition to recording the nail assignment parameters used in developing the reports, the header page also contains some assignment statistics: how many nail groups use one nail, how many groups use two nails, etc.

Samples of each report appear in Figures 8-8 to 8-11 and a description of each is presented in paragraphs 8.8.2 to 8.8.5. The monitor options associated with these reports are described in para 8.7.

For 2270 systems, a Fixture Report mode is available which produces wiring reports equivalent to the nail assignment output files .NFR and .NAR. This mode is described in para 8.8.6; a sample report is shown in Figure 8-13.

8.8.2 Nail Fixture Report (.NFR)

The Nail Fixture Report (.NFR), shown in Figure 8-8, describes how to wire the test fixture. The report specifies all connections between receiver pins and the circuit nodes. Two different versions of the report are available (selected via monitor option on the Nail Assignment page). Both versions are organized by connector, in the order in which the connectors would be wired. As shown in Figure 8-8, they differ only in the order in which the various columns appear, and the type of information in each column is defined in the figure. Component leads are listed in order of preference (i.e., preferred nodes at which to locate test nails). Digital IC output pins are listed first, then digital IC input pins or analog IC pins, followed by any pin with a name. Otherwise, the first pin in the list of components is used.



```
          2271 FIXTURE WIRE WRAP REPORT                          Page    1
                    Generated by     NASMN436
Nail Assignment Mode        NEW
.TPG Filename               SC:2720all.TPG
Report Date                 Tue 02-Aug-83 18:44:34
Report Name                 SC:2720all.NFR

Input TPG Nail Count
                                      Clock Drive      2
                                      Trigger         11
                                      Fast           139
                                      Analog          83
                                      Hybrid         813

Assigned Nail Range                             Requested Nail Range

Clock Drive             1-    2                         1-   16
Trigger                 1-   25                         1-   28
Fast                 4001- 4306                      4001- 4320
Hybrid                  1-  896                         1-  960

Assignment Statistics

    Fast Nail groups requiring one independent signal             0
    Fast Nail groups requiring two independent signals           38

    Hybrid Nail groups requiring one independent signal           1
    Hybrid Nail groups requiring two independent signals          1
    Hybrid Nail groups requiring three independent signals        2
    Hybrid Nail groups requiring four independent signals        45

Probe Nail                  F4001
Sync Nail                   F4002
-------------------------------------------------------------------------
```

Figure 8-7. Sample Nail Assignment Report Header Page.

If the SpeedPlus option is used, test nails are presented in connector order, for easy reference in wiring the test fixture. Each special (CD, CS or T) nail has 2 connection positions at the receiver interface. Nail assignment checks for NAILBIND statements in the test program to determine which hybrid or fast test nails should be wired to a special nail via an external relay position. (See Figure 8-2 for a description of this wiring.) This special wiring is also contained in the .NFR.

8.8.3 Nail Assignment Report (.NAR)

The Nail Assignment Report (.NAR), shown in Figure 8-9, is the basic report produced by Nail Assignment. It includes all the assignment information contained in the other reports, but is organized differently. It is sorted by assigned nail and thus, can be used for reference information on the actual nail assignments. In ECO or EMULATION mode, nails assigned for the first time are indicated by an *** at the beginning of that line.

If the SpeedPlus option is used, nail types are presented in the order: CD, CS, T, H and F.

8.8.4 Nail Data Base (.NDB)

The Nail Data Base (.NDB) report, shown in Figure 8-10, shows the mapping of old (temporary) nails to new (actual) nails. It is sorted by temporary nail number and thus provides an easy way to go from temporary nail to assigned nail. The report can also be edited and input to nail assignment to give the user some control over the assigning of nails.

8.8.5 Nail Assignment Listing (.NLS)

The Nail Assignment Listing (.NLS) report shown in Figure 8-11, provides information of the progress of the nail assignment function as it is performed by the system.

```
                                                                    Page  14
WIRE WRAP LISTING FOR CONNECTOR # 9: HYBRID NAILS
  SOURCE                               ASSIGNED  SOURCE                                 ASSIGNED
   .TPG    CONN      COMPONENT           NAIL      .TPG    CONN       COMPONENT           NAIL
   NAIL   NUMBER       LEAD             NUMBER     NAIL   NUMBER        LEAD             NUMBER
----------------------------------------------------------------------------------------------
|..H161...9-001.......U35.14.......H161|..H177...9-017.......U152.1........H177|
   H162   9-002       U43.14       H162|  H178   9-018       U114.6        H178|
|..H163...9-003.......U51.14.......H163|..H179...9-019.......U110.6........H179|
   H164   9-004       U59.14       H164|  H180   9-020       U117.6        H180|
|..H165...9-005.......U74.5........H165|..H181...9-021.......U73.1.........H181|
   H166   9-006       U74.12       H166|  H182   9-022       U85.4         H182|
|..H167...9-007.......U74.7........H167|..H183...9-023.......U159.8........H183|
   H168   9-008       U74.9        H168|  H184   9-024       U150.2        H184|
|..H169...9-009........Z8.9........H169|..H185...9-025.......U5.15.........H185|
   H170   9-010        Z6.5        H170|  H186   9-026       U153.2        H186|
|..H171...9-011......U173.5........H171|..H187...9-027.......U88.9.........H187|
   H172   9-012      U173.19       H172|  H188   9-028      U167.10        H188|
|..H173...9-013......U173.2........H173|..H189...9-029.......U86.4.........H189|
   H174   9-014        Z7.2        H174|  H190   9-030       Z10.2         H190|
|..H175...9-015......U119.13.......H175|..H191...9-031......U194.17........H191|
   H176   9-016      U166.18       H176|  H192   9-032       U159.6        H192|
|...GND...9-GND........................|...GND...9-GND.........................|
----------------------------------------+----------------------------------------
|..H193...9-033.......U113.6.......H193|..H209...9-049.......U65.14........H209|
   H194   9-034        Z1.4        H194|  H210   9-050       U71.14        H210|
|..H195...9-035.......U124.8.......H195|..H211...9-051.......U67.14........H211|
   H196   9-036       U69.14       H196|  H212   9-052      U168.16        H212|
|..H197...9-037.......U150.9.......H197|..H213...9-053.......U168.5........H213|
   H198   9-038      U166.17       H198|  H214   9-054      U168.19        H214|
|..H199...9-039......U132.14.......H199|..H215...9-055.......U168.2........H215|
   H200   9-040       U105.3       H200|  H216   9-056       U96.5         H216|
|..H201...9-041......U168.12.......H201|..H217...9-057......[ 2-129].......H217|
   H202   9-042       U168.9       H202|  H219   9-058      [ 2-130]       H218|
|..H203...9-043......U168.15.......H203|..H219...9-059......[ 2-131].......H219|
   H204   9-044       U168.6       H204|  H220   9-060      U121.14        H220|
|..H205...9-045.......U86.1........H205|..H221...9-061.......U183.8........H221|
   H206   9-046       U86.10       H206|  H222   9-062      U208.18        H222|
|..H207...9-047......U121.22.......H207|..H223...9-063......U147.11........H223|
   H208   9-048       U119.7       H208|  H224   9-064       U1.15         H224|
|...GND...9-GND........................|...GND...9-GND.........................|
----------------------------------------+----------------------------------------
```

where:

SOURCE .TPG NAIL	Temporary nail number defined by the user in the .CKT file (or in the .TPX file) prior to running Nail Assignment NEW mode.
CONN NUMBER	Defines the location of the connector and test pin at the receiver interface. For example, 9-017 indicates test pin 17 in connector block 9.
COMPONENT LEAD	A component lead attached to the node, taken from the %CIRCUIT section of the circuit description. For example, U15.8 meaning pin 8 of IC U15.
ASSIGNED NAIL NUMBER	The actual test fixture nail assigned to that node by the nail assignment software appearing in Test Program (.TPG).

Figure 8-8. Sample Nail Fixture ("TEMP") Report (.NFR). Page 1 of 2.

```
                                                                     Page    4

WIRE WRAP LISTING FOR CONNECTOR # 3: FAST NAILS
         ASSIGNED                                         ASSIGNED
   CONN    NAIL          NODE  COMPONENT            CONN    NAIL          NODE  COMPONENT
 NUMBER  NUMBER          NAME    LEAD             NUMBER  NUMBER          NAME    LEAD
-------------------------------------------------------------------------------------
|...3-001.F4001................PROBE.....|...3-017.F4017..........N236..U130,9....|
|   3-002 F4002                SYNC      |   3-018 F4018                          |
|**.3-003.F4003..........N786..U103,33...|...3-019.F4019...........N66..U118,34...|
|** 3-004 F4004          N784  U103,35   |   3-020 F4020                          |
|**.3-005.F4005..........N123..U145,3....|...3-021.F4021..........N104..U91,4.....|
|** 3-006 F4006          N210  U126,2    |   3-022 F4022          N243  U137,8    |
|**.3-007.F4007..........N213..U118,28...|...3-023.F4023..........N739..Z9,3......|
|** 3-008 F4008           N91  U79,7     |   3-024 F4024          N105  U91,6     |
|...3-009.F4009...........N75..U129,9....|...3-025.F4025...........N74..U129,15...|
|   3-010 F4010                          |   3-026 F4026           N79  U139,5    |
|...3-011.F4011..........................|...3-027.F4027..........N249..U130,6....|
|   3-012 F4012           N91  U79,7     |   3-028 F4028          N244  U137,9    |
|...3-013.F4013..........N240..U137,5....|...3-029.F4029..........N740..Z9,4......|
|   3-014 F4014          N239  U137,4    |   3-030 F4030          N742  U99,30    |
|...3-015.F4015..........N780..U103,41...|...3-031.F4031..........................|
|   3-016 F4016          N100  U91,7     |   3-032 F4032          N749  U99,14    |
|...3-GND...GND..........................|...3-GND...GND..........................|
-----------------------------------------+-------------------------------------------
|...3-033.F4033..........N781..U103,40...|...3-049.F4049..........................|
|   3-034 F4034                          |   3-050 F4050          N294  U120,23   |
|...3-035.F4035..........N198..U154,12...|...3-051.F4051..........N790..U103,43...|
|   3-036 F4036          N789  U103,44   |   3-052 F4052           N90  U79,9     |
|...3-037.F4037...........N90..U79,9.....|...3-053.F4053..........N219..[ 2-145]..|
|   3-038 F4038          N245  U130,2    |** 3-054 F4054          N785  U103,34   |
|**.3-039.F4039...........N85..U103,32...|**.3-055.F4055..........N240..U137,5....|
|** 3-040 F4040           N75  U129,9    |** 3-056 F4056          N236  U130,9    |
|...3-041.F4041..........N115..U156,23...|...3-057.F4057..........N786..U103,33...|
|   3-042 F4042          N329  U99,13    |   3-058 F4058           N65  U118,33   |
|...3-043.F4043...........N78..U139,19...|...3-059.F4059..........N217..U156,15...|
|   3-044 F4044          N312  Z10,9     |   3-060 F4060          N806  U118,19   |
|...3-045.F4045...........N86..U103,46...|...3-061.F4061..........N425..U96,9.....|
|   3-046 F4046                          |   3-062 F4062          N783  U103,37   |
|**.3-047.F4047..........N779..U103,42...|**.3-063.F4063..........N240..U137,5....|
|** 3-048 F4048           N91  U79,7     |** 3-064 F4064           N66  U118,34   |
|...3-GND...GND..........................|...3-GND...GND..........................|
-----------------------------------------+-------------------------------------------
```

where:

NODE NAME	Node name assigned to the schematic and in the %CIRCUIT section of the circuit description.

Other headers are defined on sheet 1.

** indicates new wires assigned in ECO mode.

Figure 8-8. Sample Nail Fixture ("FINAL") Report (.NFR). Page 2 of 2.

```
    ASSIGNED          TEMP                  Page: 2
      NAIL  NODE      NAIL  CONNECTOR    COMPONENTS
-----------------------------------------------------------------
       CD1  N111       CD1  2-070     *U108.6, *U166.1, U101.2, U103.2,
                                      U108.2, U118.15, U134.11, U138.24,
                                      U146.9, U148.11, U155.11, U156.1,
                                      U164.1, U165.11, U171.8, U178.1,
                                      U192.11, U205.1

       CD2  N201       CD5  2-071     *U109.8, U96.11, U99.32, U109.12

        T1  N254        T1  2-117     *U138.2, *Z3.2, U135.4, U135.9

        T2  N117        T2  2-118     *U118.6, *U131.20, *Z4.2, U105.15,
                                      U126.11, U133.2, U135.5, U136.2,
                                      U159.1, U159.4

        T5  N252        T5  2-125     *U138.3, *Z4.7, U129.11, U139.11

        T6  N220        T6  2-126     *U118.7, *Z3.5, U135.12, U138.17

        T9  N219        T9  2-133     *U99.9, *U138.15, *Z9.9, U107.4

       T10  N221       T10  2-134     *U118.8, *Z3.6, U135.2, U169.4

       T13  N235       T13  2-141     *U138.16, *Z4.6, U130.19, U137.19

       T14  N807       T14  2-142     *U118.20

       T17  N232       T17  2-149     *U138.19, *Z3.7

       T21  N233       T21  2-157     *U138.20, *Z3.8

       T25  N234       T25  2-159     *U138.21, *Z3.9

        H1  N390        H2 10-001     *U164.20, U121.10, U121.13, U132.10,
                                      U132.13, U136.4, U136.10, U141.10,
                                      U141.13, U145.13, U178.7, U179.13,
                                      U190.19
```

where:

ASSIGNED NAIL	The actual test fixture nail assigned to that node by the nail assignment software, and appearing in the test program. Three asterisks (***) preceding the assigned nail indicates a new nail assignment.
NODE	Node name assigned in the %CIRCUIT section of the circuit description. If there is no IDD input file, no entry appears in this column.
TEMP NAIL	Temporary nail specified by the user in the %ADAPTOR:2272 section of the circuit description (.CKT) file. For example, in the following CDL (Circuit Description Language) statement: H9=U1.1; the temporary nail number is 9.
CONNECTOR	Defines the location of the connector and test pin at the receiver interface. For example, 10-017 indicates test pin 17 located in connector block 10.
COMPONENTS	Component leads attached to the node, taken from the %CIRCUIT section of the circuit description. For example, U68.3,U67.4 An asterisk (*) indicates a digital output pin. Components appear in a preferred-component order, which minimizes ECO adjustments and simplifies locating the nails. If there is no input .IDD file, no output will appear in this column.

Figure 8-9. Sample Nail Assignment Report (.NAR).

```
/*
 TEMP NAIL NUMBER -> ASSIGNED NAIL NUMBER                Page: 2
------------------------------------------------------------
*/
%NAILMAP 2271 ECO;
     CD1->    CD1;           CD2->    CD5;
      T1->     T2;            T2->     T6;           T3->     T10;
      T4->    T14;            T5->     T1;           T6->      T5;
      T7->     T9;            T8->    T13;           T9->     T17;
     T10->    T21;           T11->    T25;
   PROBE-> F4001;         SYNC-> F4002;              H3->    H710;
      H4->   H721;            H5->    H13;           H6->     H27;
      H7->    H28;            H8->    H44;           H9->     H45;
     H10->    H46;           H11->    H47;          H12->     H54;
     H13->   H683;           H14->    H55;          H15->     H56;
     H16->    H57;           H17->   H251;          H18->    H300;
     H19->   H185;           H20->   H224;          H21->    H113;
     H22->   H114;           H23->   H115;          H24->    H116;
     H25->   H129;           H26->   H130;          H27->    H131;
     H28->   H132;           H29->   H145;          H29->    H822;
     H30->   H146;           H31->   H147;          H32->    H148;
     H33->   H823;           H33->   H161;          H34->    H837;
     H34->   H162;           H35->   H838;          H35->    H163;
     H36->   H164;           H36->   H839;          H37->    H817;
     H37->     H9;           H38->    H10;          H39->     H11;
     H40->    H68;           H40->   H833;          H41->     H69;
     H41->   H834;           H42->   H835;          H42->     H70;
     H43->    H81;           H43->   H849;          H44->    H850;
     H44->    H82;           H45->    H16;          H46->     H74;
     H47->    H83;           H48->    H84;          H49->     H97;
     H49->   H821;           H50->    H98;          H51->     H99;
     H52->   H100;           H53->   H526;          H54->    H527;
     H55->   H502;           H56->   H503;          H57->    H210;
     H58->   H211;           H59->   H196;          H60->    H209;
     H80->   H120;           H81->   H121;          H82->    H134;
     H83->   H135;           H84->   H774;         H106->    H819;
    H106->    H33;          H107->    H18;         H108->     H34;
    H109->   H207;          H111->   H814;         H111->      H1;
    H112->    H63;          H113->    H53;         H114->    H125;
    H116->    H40;          H117->    H21;         H118->     H58;
    H119->    H24;          H120->    H37;         H121->    H421;
    H122->   H422;          H125->   H225;         H126->    H226;
    H127->   H128;          H128->   H143;         H129->    H133;
    H130->   H235;          H131->   H122;         H132->    H236;
    H133->   H301;          H134->   H329;         H135->    H177;
    H136->   H155;          H137->   H656;         H138->    H657;
    H139->   H658;          H140->   H659;         H141->    H194;
    H142->   H451;          H143->   H758;         H144->    H711;
    H145->   H712;          H146->   H346;         H147->    H644;
    H150->    H39;          H151->   H247;         H152->    H105;
    H153->   H747;          H154->   H748;         H155->    H636;
```

where:

TEMP NAIL NUMBER	Temporary nail specified by the user in the %ADAPTOR:2272 section of the circuit description.
ASSIGNED NAIL NUMBER	The actual test fixture nail assigned to that node by the nail assignment software.

If the user is creating a new .NDB for use by the nail assignment software, one or more of the following headers must be put in the new file as applicable:

%NAILMAP 2272 NEW; for NEW mode
%NAILMAP 2272 ECO; for ECO mode
%NAILMAP 2270 EMULATION; for emulation mode

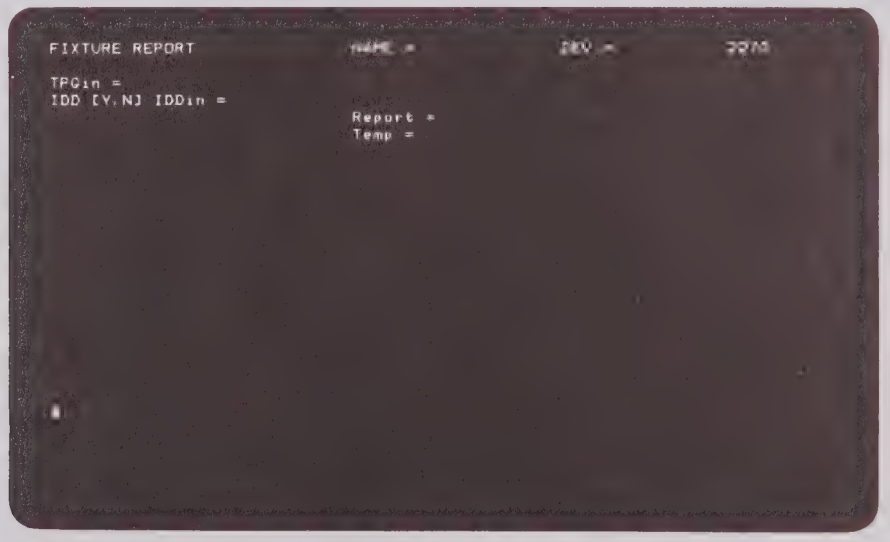
Figure 8-10. Sample Nail Data Base (.NDB).

```
                    2271 NAIL ASSIGNMENT IN PROGRESS

                            Target Machine: 2271
                            Software ID:    NASMN275 A0
                            Mode:   NEW
                            Board Name:     R72RLYBD
                            Date:   Mon 23-Aug-82 10:26:50

Process:            Copying Input TPG File
                    Process completed:     0 errors detected
                                           0 warnings detected

Process:            Creating Nail Record File (.VPN)
                    Process completed:     0 errors detected
                                           0 warnings detected

Process:            Creating Nail Assignment File (.VPR)
                    Process completed:     0 errors detected
                                           0 warnings detected

Process:            Mapping old nails to new
                    Process completed:     0 errors detected
                                           0 warnings detected
                           |                     |
                           |                     |
                           |                     |

                              NAIL ASSIGNMENT COMPLETED

                              Cumulative errors:      0

                              Cumulative warnings:    0
```

Figure 8-11. Sample Nail Assignment Listing (.NLS).

8.9 2270 FIXTURE REPORTS

The 2270 Fixture Report monitor mode options are shown in Figure 8-12.



```
FIXTURE REPORT                  NAME =            DEV =          2270

TPGin =
IDD [Y,N] IDDin =
                                Report =
                                Temp =
```

Figure 8-12. 2270 Fixture Report Mode Options Page.

TPGin=
Supplies the device and name of the source test program to the Fixture Report mode if they differ from the "DEV =" and "NAME =" options.

IDD [Y,N] IDDin=

[Y] When Y is specified the "IDDin =" file spec option appears on the page, allowing the user to specify an alternate device and name for the

In-circuit Diagnostic Data file (.IDD). The device and name default to the "DEV =" and "NAME =" option entries from the top of the page.

[N] N specifies no input .IDD file.

Report =
Specifies the device on which to store the fixture wiring report files (.NFR, .NAR and .NLS). Device defaults to the "DEV =" device.

Temp=
Specifies the device on which to store the temporary files created by Fixture Report. Defaults to the "DEV =" device.

The Nail Fixture Report (.NFR file) for a 2270 system is shown in Figure 8-13. It lists the following connection information:

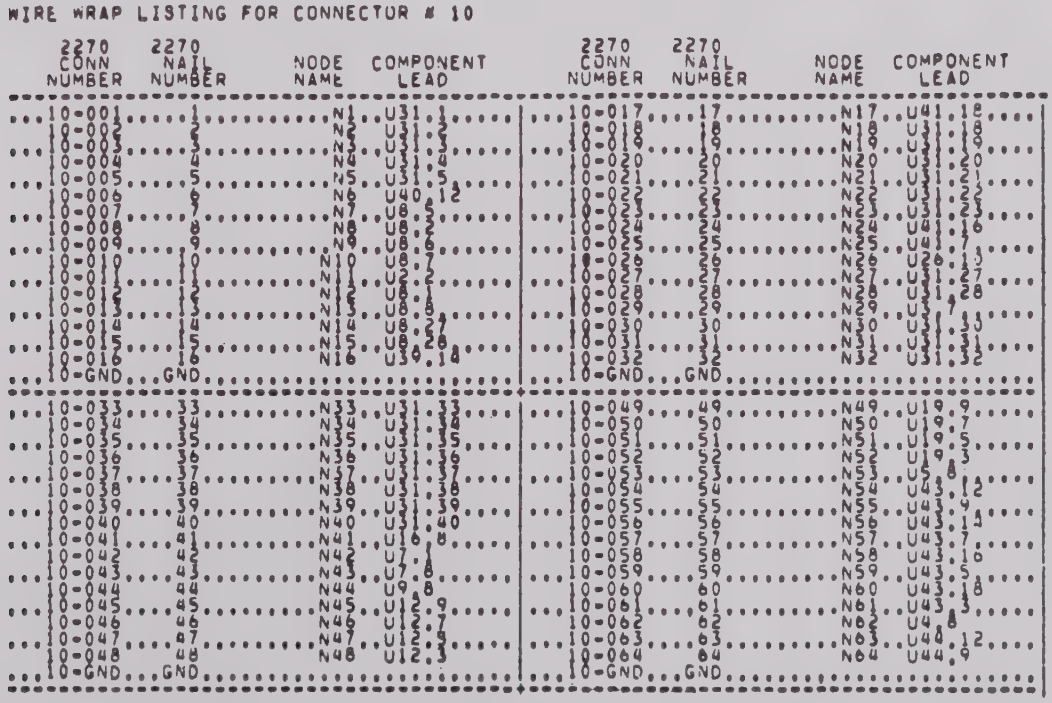
1. The connection at the receiver interface. For example, 10-042 indicates pin 42 on block 10.

2. The system test nail number.

3. The node name assigned by the user in the %CIRCUIT section of the circuit description.

4. A component lead attached to each node.

A Nail Assignment Report (.NAR) is also available, which provides connection information on each system nail, presented sequentially by nail number for user convenience.



```
WIRE WRAP LISTING FOR CONNECTOR # 10

   2270    2270                                    2270    2270
   CONN    NAIL       NODE  COMPONENT              CONN    NAIL        NODE  COMPONENT
  NUMBER  NUMBER      NAME    LEAD                NUMBER  NUMBER       NAME    LEAD
-----------------------------------------------------------------------------------------
...10-001.....1..........N1..U31.1.....|...10-017....17..........N17..U41.18....|
   10-002     2          N2  U31.2     |   10-018    18          N18  U31.18    |
...10-003.....3..........N3..U31.3.....|...10-019....19..........N19..U31.19....|
   10-004     4          N4  U31.4     |   10-020    20          N20  U31.20    |
...10-005.....5..........N5..U31.5.....|...10-021....21..........N21..U31.21....|
   10-006     6          N6  U40.12    |   10-022    22          N22  U31.22    |
...10-007.....7..........N7..U8.5......|...10-023....23..........N23..U31.23....|
   10-008     8          N8  U8.2      |   10-024    24          N24  U41.16    |
...10-009.....9..........N9..U8.6......|...10-025....25..........N25..U41.7.....|
   10-010    10         N10  U8.7      |   10-026    26          N26  U26.13    |
...10-011....11.........N11..U2.2......|...10-027....27..........N27..U31.27....|
   10-012    12         N12  U8.1      |   10-028    28          N28  U31.28    |
...10-013....13.........N13..U8.8......|...10-029....29..........N29..U31.7.....|
   10-014    14         N14  U8.27     |   10-030    30          N30  U31.33    |
...10-015....15.........N15..U8.26.....|...10-031....31..........N31..U31.31....|
   10-016    16         N16  U39.14    |   10-032    32          N32  U31.32    |
...10-GND...GND........................|...10-GND...GND.........................|
---------------------------------------+----------------------------------------|
...10-033....33.........N33..U31.33....|...10-049....49..........N49..U19.9.....|
   10-034    34         N34  U31.34    |   10-050    50          N50  U19.7     |
...10-035....35.........N35..U31.35....|...10-051....51..........N51..U19.5.....|
   10-036    36         N36  U31.36    |   10-052    52          N52  U19.3     |
...10-037....37.........N37..U31.37....|...10-053....53..........N53..U5.8......|
   10-038    38         N38  U31.38    |   10-054    54          N54  U43.12    |
...10-039....39.........N39..U31.39....|...10-055....55..........N55..U43.9.....|
   10-040    40         N40  U31.40    |   10-056    56          N56  U43.13    |
...10-041....41.........N41..U16.8.....|...10-057....57..........N57..U43.7.....|
   10-042    42         N42  U7.1      |   10-058    58          N58  U43.16    |
...10-043....43.........N43..U7.8......|...10-059....59..........N59..U43.5.....|
   10-044    44         N44  U9.8      |   10-060    60          N60  U43.18    |
...10-045....45.........N45..U12.9.....|...10-061....61..........N61..U43.3.....|
   10-046    46         N46  U12.7     |   10-062    62          N62  U4.8      |
...10-047....47.........N47..U12.5.....|...10-063....63..........N63..U40.12....|
   10-048    48         N48  U12.3     |   10-064    64          N64  U44.9     |
...10-GND...GND........................|...10-GND...GND.........................|
---------------------------------------+----------------------------------------|
```

Figure 8-13. Sample 2270 Nail Fixture Report.

Debugging the Test Program – Section 9

9.1 GENERAL

The basic task used for creating an object file is the translator task, which uses the source test program (.TPG) as the input. Additional tasks are generally required to correct errors. Program format (syntax) errors are found and reported by the translator. Program logic errors are found when the program is run to test an actual pc board. This section deals mainly with debugging procedures that can be used to create an error-free object file. Figure 9-1 illustrates the steps used in translating and debugging a test program.

9.2 TRANSLATING THE TEST PROGRAM

9.2.1 General

Run the translator task by requesting the Translate mode options page, assuring that all file information is correct for each option, and pressing the RUN key. The translator creates, in addition to the object file, an error list (.LIS) file. As the statements in the .TPG are translated, they are also checked for language structure (syntax) and for conflicts in assignments. The error messages include the statement at which the error was recognized. A caret is inserted under the point of error recognition. The location and type of error are also listed below the statement. Example:

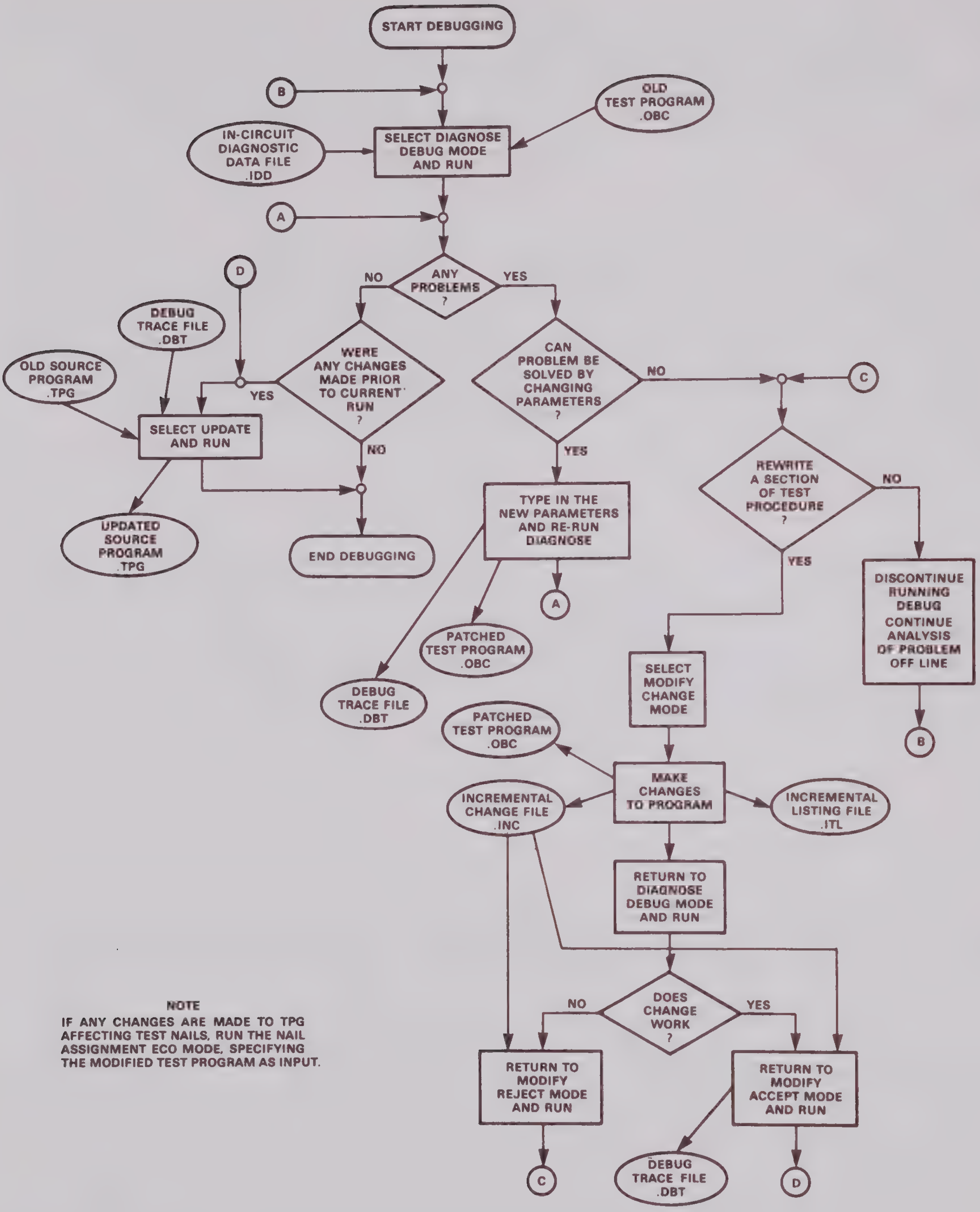
```
MEAS R INTO R1 DYL=.005 V=5.0 [BRANCH H100];
                  ^
ERROR NO.  2 AT LINE 2:
SYNTAX ERROR
```

The keyword DLY was spelled incorrectly in the measure statement. This error is classed as a syntax error and the message indicates that it is the second error that was found and the statement was the second line in the file.

The caret points to the place where the translator recognized that an error exists. This may not be the point at which the error actually occurred; some analysis of the indication and message may be required to find the error. A list of translator error-type messages appears in Appendix B of this manual.

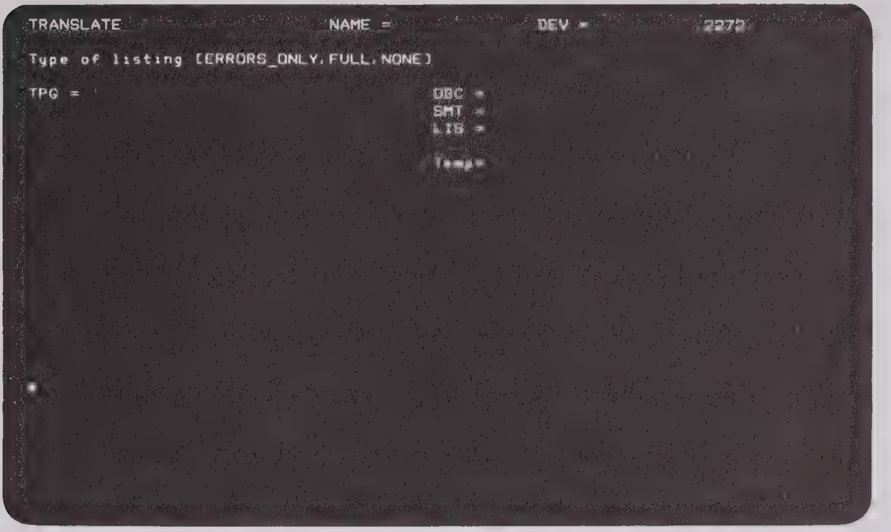
9.2.2 Translate Mode

The Translator is the monitor mode that converts a readable source language test program into two machine-readable binary files called the Object Code file (.OBC) and the Symbol Table file (.SMT). The binary .OBC file is used by the Diagnose mode for board testing. The .SMT file contains pointers to labels and variables that are used in debugging. Test Language statements are checked during translation, and suitable messages are delivered if errors are detected. The translator indicates what part of an erroneous statement is not acceptable. The source language test program can then be edited to correct the error and, when the editor is exited, the translator can again be used to translate the test program. This process can be repeated until an error-free program is achieved and the .OBC and .SMT files are produced.


START DEBUGGING
B
OLD TEST PROGRAM .OBC
IN-CIRCUIT DIAGNOSTIC DATA FILE .IDD
SELECT DIAGNOSE DEBUG MODE AND RUN
A
ANY PROBLEMS ?
NO
YES
D
DEBUG TRACE FILE .DBT
OLD SOURCE PROGRAM .TPG
WERE ANY CHANGES MADE PRIOR TO CURRENT RUN ?
YES
NO
SELECT UPDATE AND RUN
UPDATED SOURCE PROGRAM .TPG
END DEBUGGING
CAN PROBLEM BE SOLVED BY CHANGING PARAMETERS ?
NO
YES
C
TYPE IN THE NEW PARAMETERS AND RE-RUN DIAGNOSE
A
PATCHED TEST PROGRAM .OBC
DEBUG TRACE FILE .DBT
REWRITE A SECTION OF TEST PROCEDURE ?
NO
YES
DISCONTINUE RUNNING DEBUG CONTINUE ANALYSIS OF PROBLEM OFF LINE
B
SELECT MODIFY CHANGE MODE
PATCHED TEST PROGRAM .OBC
MAKE CHANGES TO PROGRAM
INCREMENTAL CHANGE FILE .INC
INCREMENTAL LISTING FILE .ITL
RETURN TO DIAGNOSE DEBUG MODE AND RUN
DOES CHANGE WORK ?
NO
YES
RETURN TO MODIFY REJECT MODE AND RUN
RETURN TO MODIFY ACCEPT MODE AND RUN
C
DEBUG TRACE FILE .DBT
D

NOTE
IF ANY CHANGES ARE MADE TO TPG AFFECTING TEST NAILS, RUN THE NAIL ASSIGNMENT ECO MODE, SPECIFYING THE MODIFIED TEST PROGRAM AS INPUT.

Figure 9-1. Debugging Process.



```
TRANSLATE                    NAME =               DEV =          2272
Type of listing [ERRORS_ONLY,FULL,NONE]
TPG =                                   OBC =
                                        SMT =
                                        LIS =

                                        Temp=
```

Figure 9-2. Translate Mode Options Page.

Typing TR from the command line of any monitor page, except the Diagnose page where TRAN must be typed, will call Translate mode and the page shown in Figure 9-2 is displayed.

Type of listing [ERRORS_ONLY,FULL,NONE]

[ERRORS_ONLY] Selecting ERRORS_ONLY writes only test program statements which contain test language errors to the listing file.

[FULL] Selecting FULL directs the translator to write a complete listing file containing all statements from the test program (.TPG) file, whether or not the statements contain errors.

[NONE] No listing (.LIS) file is produced when NONE is specified.

TPG =

An alternate device, name, and/or extension to use for the test language program file can be specified with this option. If no response is filled-in for this option, the "DEV =" and "NAME =" options are used for the device and name, and .TPG is used as the extension name.

OBC =

An alternate device, name, and/or extension for the resulting binary test program can be supplied in this option. The defaults for device and name are the "DEV =" and "NAME =" options, and the extension defaults to .OBC.

SMT =

An alternate device, name, and/or extension for the symbol table file can be specified in this option. If no entry is made for this option, the defaults are the "DEV =" option, "NAME =" option, and .SMT, respectively.

LIS =

System diagnostic and error messages are sent to the Listing file (.LIS) or device. If no entry is made here, the file specification defaults to the "DEV =" and "NAME =" entries and the extension .LIS. The video display can be used as the listing device by specifying LIS = TI:, but no file is created for later examination.

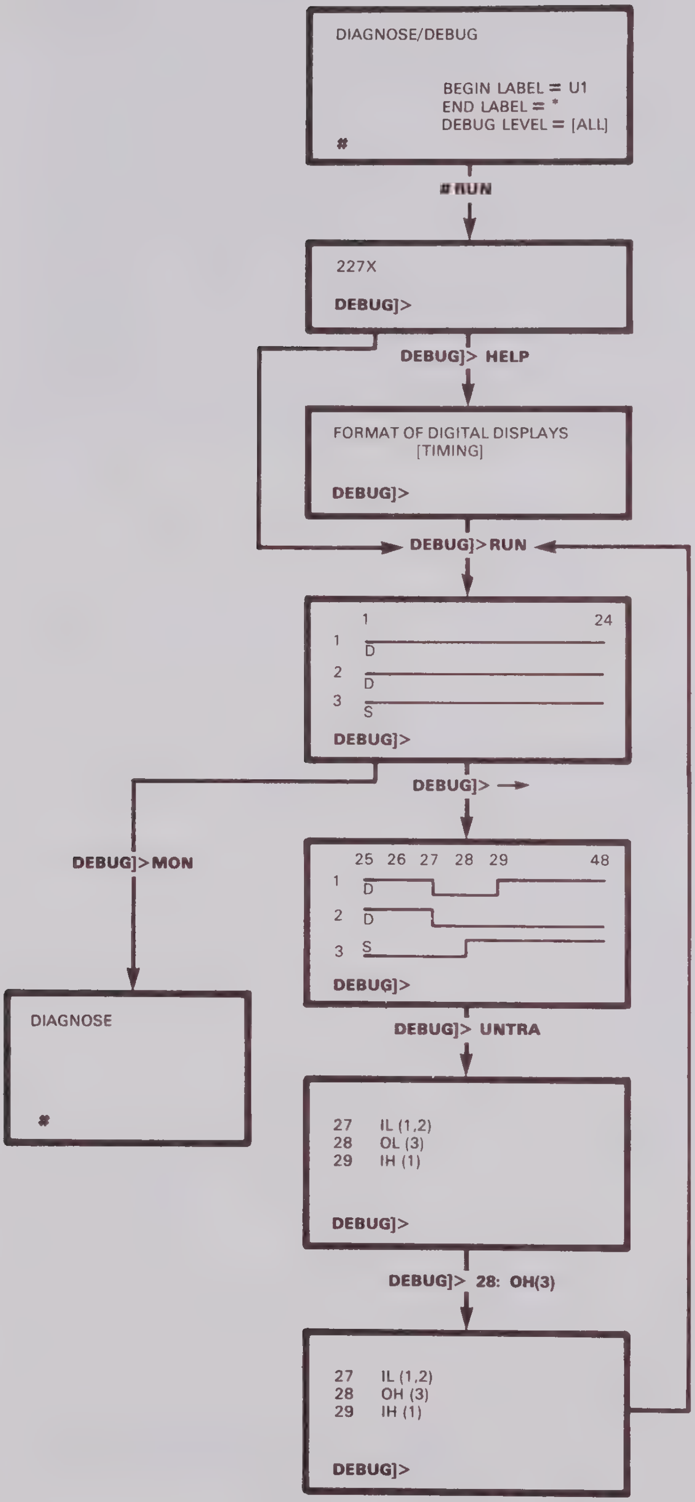

DIAGNOSE/DEBUG
BEGIN LABEL = U1
END LABEL = *
DEBUG LEVEL = [ALL]
#
#RUN
227X
DEBUG]>
DEBUG]> HELP
FORMAT OF DIGITAL DISPLAYS
[TIMING]
DEBUG]>
DEBUG]>RUN
1
24
1
D
2
D
3
S
DEBUG]>
DEBUG]> →
DEBUG]>MON
25 26 27 28 29
48
1
D
2
D
3
S
DEBUG]>
DIAGNOSE
#
DEBUG]> UNTRA
27 IL (1,2)
28 OL (3)
29 IH (1)
DEBUG]>
DEBUG]> 28: OH(3)
27 IL (1,2)
28 OH (3)
29 IH (1)
DEBUG]>

Figure 9-3. Typical Debug Display.

Temp =

Indicates to the translator which device to use for storage of temporary work files created during the translation process. The default, if no entry is made here, is the "DEV =" option.

Example

Type TR from any monitor page (TRAN from Diagnose page) command line. Press the RUN key or type RUN. The .TPG file specified by the "NAME =" and "DEV =" entries in the monitor page is translated. If any errors are detected, the lines containing them are displayed on the terminal. If translation is successful, a .OBC file results that can be run from the Diagnose options page.

9.2.3 Correcting Errors Resulting from Translator Task

Corrections to the test program are made by requesting the Editor mode and running it with the .TPG file. The translator is then run on the new version of the .TPG file.

To locate errors easily, it may be advisable to select the option on the Translate page that will cause a complete TPG listing to appear in the .LIS file. Then print the file for easy reference and marking. This may be important for .TPG files which have significant user modifications from the ATG-produced version. When errors have been corrected, this option should be turned off to reduce translator run time. Use the video display to list any remaining or newly introduced errors. This display is selected on the options page.

9.3 DEBUGGING THE TEST PROGRAM

9.3.1 General

Once an object file has been created with no translation errors, the task of debugging the program for logical errors can begin. Select the Diagnose options page, specify the DEBUG mode, assure that all appropriate options are properly specified, and initiate the running of this mode. Refer to para 9.4 for a complete description of the Diagnose and Debug options. Figure 9-3 illustrates the process of stepping through the various debug displays and the issuing of debug commands. The begin and end labels define the component or group of components for which the debug features will be active during the test run. Mount a known good board on the test fixture and type RUN to start debugging. It is a good idea to have checked out the fixture, using debug mode, prior to debugging the program. Refer to para 9.3.7 for fixture debugging.

The features of Debug are detailed test result data and ease of making, and testing, changes to the test program. When in the Debug sub-mode, the diagnose task (Run-Time System) tests the pc board, and data resulting from the test are displayed on the video display. The user can then analyze the displayed data, instruct the test system to make changes to the test program, and re-run the program.

The set up of the Diagnose options page for running the debug mode differs from that for running a test. A few added options appear on the page when the DEBUG mode is selected; these are shown in the top block of Figure 9-3. When debugging a program, the user frequently runs one component test at a time. Therefore, options are provided which allow for identifying a specific test. The start of each test in a program is noted by a label at the first statement of the test. This label would be inserted on the options page by moving the cursor to the "Begin label =" option line and typing in the label name. The end of one test is the same as the beginning of the next test and, therefore, the next label. A special character, the asterisk (*), is used to specify the next label. For example, the test burst for U1 would be specified on the options page by filling in the label options

Begin label = U1
End label = *

A good way to begin program debugging is to leave the "Begin label =" option unspecified (null), which directs the system to start debugging at the beginning of the test program, and also to leave the "End label =" option null, which directs the system to debug the whole test program. Set the "Debug level" to FAILURES, and run debug. The whole program will be run, but only the results of tests which fail will be displayed on the screen.

The "Debug level" options selects the amount of test reporting which is displayed following a debug run. The default setting, FAILURES, displays only the results of tests for components which failed. Enabling ALL displays all test statements in the debug range, whether or not they failed.

Pressing RUN at the Diagnose page prompt, (#), displays the Debug mode prompt illustrated in the second block of Figure 9-3. At this point, additional Debug option selections can be made by pressing HELP, and the sub-mode display appears. The option of prime importance provides for selecting the form of digital result data, Timing or Errors.

When RUN is again pressed, the test is performed and the results displayed. Figure 9-3 shows that the timing diagram first displayed did not contain the steps of interest in the test. To display the next group of steps, the right arrow key is pressed. To see the program statements for the displayed steps of the diagram, the UNTRAnslate command is entered. Changes can then be made and the test re-run.

9.3.2 Analog Component Debugging

When an analog component is tested in debug mode, a display of the test statements and test results appears. The debug prompt also appears at the bottom of the display. At this point, the user may enter any of the debug commands appropriate to analog tests. Test statements can be modified in any one of the following ways:

1. Change the value of a parameter in a statement.
2. Add a parameter to a statement.
3. Add nails to a nail list.
4. Delete a statement.
5. Delete nails from a nail list.
6. Delete a parameter from a statement.
7. Change to a 4-terminal guarded measurement.
8. Restore the test to its original state.
9. Swap source and measure nodes (CHA and CHB).

Other analog types of program changes must be done using the Modify task described in para 9.3.5. Refer to para 9.4.2 for detailed descriptions of commands that can be used in debug.

To change a parameter value in a statement, type the statement number, as shown in the debug display, followed by the parameter name and new value. This is shown in the following example.

Example

Debug the test for CR1. All test program statements are to be listed and the results of all measurements are to be listed.

Required Options: Select "Debug level" to be ALL, "Begin label = CR1", and "End label = *". The display appears as follows:

Display:

```
CR1:

  1: SET SCAN AT (CHA=240:CHB=241);
  2: SET DCI DCM DLY=30.5U RDLY=30.5U FORCE;
  3: SET DCV DCS V=1.7 DLY=0.0 I=20M;
  4: MEAS DCV DCM INTO VVAL1 HI=1.0 LO=565.0M DLY=30.5U
          RDLY=30.5U
  5:    [ BRANCH OKCR1; ];
  6: SET DCI DCS I=10.0M DLY=0.0;
  7: MEAS DCV DCM INTO VAR HI=600.0M LO=200.0M DLY=30.5U
          RDLY=30.5U;
  8: SET DCI DCS I=0.0 DLY=0.0;
  9: SET DCV DCS V=0.0 DLY=0.0 I=20M;
 10: TEST ARITH VAR HI=10.0 LO=0.0;

                Stmt   Type        Last Meas Cmp

                 4:     V            0.70  PASS
                 7:     V             .66  HIGH
                10:     AR           8.21  PASS

DEBUG]>
```

Meaning: The value for V in statement 4 measured 0.7 V, which passed. The value for V in statement 7 measured .66 V, which was high. The value for AR in statement 10 measured 8.21 ohms, which passed.

After analyzing the circuit, it is decided that the high value of voltage VAR should have been specified in statement 7: as HI=782.0M. To make this change, type

```
DEBUG]> 7: HI=782.0M
```

The new parameters will be interpreted and patched into the .OBC file. Now type

```
DEBUG]> RER
```

and the test will be re-run with the new data.

9.3.3 Digital Component Debugging

When a digital component is tested in debug mode, a display of the driven and sensed logic states appears. The debug prompt also appears at the bottom of the display. At this point the user may enter any of the debug commands appropriate to digital tests. The test statements that correspond to the steps in the logic state display can be shown on the video display with the untranslate command. Test statements can be modified in any of the following ways:

1. Add a parameter to a statement.
2. Add a statement.
3. Add nails to a nail list.
4. Delete a statement.
5. Delete nails from a nail list.
6. Delete a parameter from a statement.
7. Include all learned states.
8. Restore test to original state.
9. Insert single-step (not incremental) state parameters into a statement.
10. Change a SET LOGIC statement to a TEST LOGIC statement.
11. Change a TEST LOGIC statement to a SET LOGIC statement.
12. Change burst parameters such as ITT and IST.
13. Add or delete pull-up resistors.

Other types of program changes must be made with the Modify task described in para 9.3.5. Refer to para 9.4.2 for detailed descriptions of commands that can be used in Debug.

The following example shows the process of making a change to a logic state that was incorrect in the test procedure.

Example

Debug the procedure for IC1, a 7400 on the board named LOGIC. Show a timing diagram for all test steps and all active pins.

Options selected on Diagnose page:

"Mode" is DEBUG
"Begin label =" is IC1
"End label =" is *
"Debug level" is ALL

The Debug sub-mode display option:

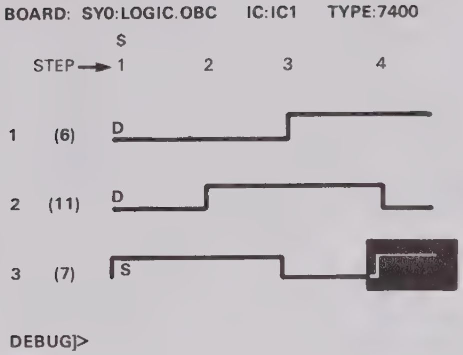
"Format of digital displays" is TIMING

Assuming that there was an error at output pin 3 at step 4, the debug display is as shown in Figure 9-4.

Meaning: Pin 3 is sensed high in step 4. Step 4 trace of pin 3 is shown in reverse video to show that it is in error because it was not expected to go high when pin 2 went low.

Type UNTRA to display the test program statements that correspond to the steps of the digital display.

```
    1: IC(6,11) IL(6,11) OS(7) OH(7);
    2: IH(11);
    3: IH(6) OL(7)
    4: IL(11);
DEBUG]>
```



```
BOARD: SY0:LOGIC.OBC    IC:IC1    TYPE:7400

DEBUG]>
```

Figure 9-4. Digital Debug Display.

When the debug timing display and the test statements are analyzed, it becomes evident that the timing display shows exactly what should have happened for a good test of the IC. Test statement 4 is in error because it should have indicated that the output would go high. Correct statement 4 by typing the following debug commands:

```
4: OH(7);
```

The debug displays may be copied on the strip printer. In the case of the digital example with TIMING selected, the printout will have HHH in place of reverse video for error state in step 4 for pin 3. LLL, MMM or HHH means that a low, a mid-state, or a high was sensed which was not the expected state.

A change may be made to any of the statements in the debug group by first typing UNTRA to display the debug data for this test then typing the statement number shown on the display, followed by the variable name and new value that is to be inserted. This is typed in response to the "DEBUG]>" prompt. This change will be interpreted and patched and conditionally inserted in the .OBC file. It will also be saved in the .DBT file for final disposition in the .OBC file and for inclusion in the .TPG file after debugging is completed.

A change can be tried by typing RER in response to the DEBUG]> prompt to rerun the burst. If the change is unacceptable, it can be deleted by typing REM in response to the DEBUG]> prompt. If the change is acceptable, debugging can continue to the next test by typing RUN. Other debug commands that may be used are described in para 9.4.2.

9.3.4 SpeedPlus Option Debugging

Test workstations that have the SpeedPlus optional hardware provide capability for testing digital devices at operational speed. These workstations allow for as many as 4 clock drive signals and a synchronizing signal. A burst may consist of sections that are executed at the normal driver/sensor test rate and sections that are executed at a fast clock rate. The digital debug timing display indicates the normal and fast portions of the burst by displaying a column of up arrows for a transition to fast mode and a column of down arrows for a transition to normal mode. Figure 9-5 shows a display as it would be printed on the line printer, with fast transitions. Steps 7 through 11 are executed at the fast rate. This display shows that device pin 6 at fast nail 4050 was sensed incorrectly at steps 7, 9, and 10 of the high-speed portion of the burst.

```
Board : DM3:7400F.OBC          IC : U25    Type: 7400F

                       ^                 v
  STEP->    2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 End
           _
1    (1)  /D \_____ . ^^^ .  .  .  .  . vvv .  .  .  .  .  .  .  .  .  .
           _____
2    (2)  /D    \__ . ^^^ .  .  .  .  . vvv .  .  .  .  .  .  .  .  .  .
              _____
3    (3)  _S_/      . ^^^ .  .  .  .  . vvv .  .  .  .  .  .  .  .  .  .
                          _____
4 (4042)   .  .  .  . ^^^/D    \_____ . vvv .  .  .  .  .  .  .  .  .  .
                             _____
5 (4049)   .  .  .  . ^^^_D_/     \__ . vvv .  .  .  .  .  .  .  .  .  .

6 (4050)   .  .  .  . ^^^_S__________ . vvv .  .  .  .  .  .  .  .  .  .
                         ^^^   ^^^^^^

                                            _     _____
8   (20)   .  .  .  . ^^^ .  .  .  .  . vvv/S \__/      .  .  .  .  .  .
                                            _____
9 (4065)   .  .  .  . ^^^ .  .  .  .  . vvv/D    \_____ .  .  .  .  .  .
                                               _____
10  (34)   .  .  .  . ^^^ .  .  .  .  . vvv_D_/     \__ .  .  .  .  .  .
                                                           _     _____
11  (35)   .  .  .  . ^^^ .  .  .  .  . vvv .  .  .  .  . /S \__/       .
                                                           _____
12  (36)   .  .  .  . ^^^ .  .  .  .  . vvv .  .  .  .  . /D    \_____ .
                                                              _____
13  (49)   .  .  .  . ^^^ .  .  .  .  . vvv .  .  .  .  . _D_/     \__ .
```

Figure 9-5. Fast Digital Debug Display.

In addition to the driver/sensor timing display, a clock timing display can be produced which shows all the clock drive signals, the clock sync signals and the driver/sensor timing list. The clock timing display (example shown in Figure 9-6) is presented when the command 'CLOCK' is issued. The Burst timing display is returned when the command 'TIMING' is issued.

Example:

Display the current clock timing diagram (Figure 9-6) and change the driver/sensor timing list.

```
U1:     /* Pass */
  1:     SET  CLOCK AT(CDA=0:CDB=0:CDC=0:CDD=0:CSYNC=1)
         TINC=41.67N         NINC=6         DST(D2,T4,S0)
         SIGNAL(CDA=0:CDB=0:CDC=0:CDD=0:CSYNC=L0,H2,L3,H5)
         VCDH=4.00 VCDL=200.00M VCST=1.50  LOSLEW OFF
==========================================================================
F=4.00MEGhz    PERIOD=250.02N   (sync)   TINC=41.67N   NINC=6  PAGE: 1 OF 1
      0n         42n        83n        125n       167n       208n       250n
      +----------+----------+----------+----------+----------+----------+

                             __________                       __________
(CS1 )\_____________________/          \_____________________/          \
DST   S                     ED                    T
      0          1          2          3          4          5          0
;
```

Figure 9-6. Clock Debug Display.

9.3.5 Modify

If editing the test program involves more than changing parameters in existing analog statements or adding or deleting statements in a digital burst, the Modify task must be used. The Modify task allows specified sections of a test program to be edited, translated and inserted into the .OBC file without changing the .TPG and without translating the whole program.

To modify a section of a program, the Modify option page is called and the change mode is selected. The begin and end labels are entered and the Change mode is run.

Change mode results in a separate .INC file which is presented for editing. This .INC file contains the segment of the test program bounded by the specified labels, and other data necesary for independent translation (e.g., applicable ASSIGN LGC statement). On initial runs of change, this INC file is newly created by extracting text from the .TPG; if an .INC exists, that existing file is used. When editing has completed, the Translator executes in a special mode with the specified data set, which consists of the .INC file, a copy of the .OBC file and a copy of the .SMT file. These copies of the .OBC and .SMT files are modified according to the .INC file, and unless translation errors occur, are ready for subsequent debugging/testing. If translation errors occur (when the .INC has syntax errors), the partially changed .OBC and .SMT files are deleted, and subsequent Modify/Change runs will recopy the original .OBC and .SMT. Note the caution below with respect to OBC/SMT consistency. When the translator runs, it produces an .ITL error listing file.

LIST mode of MODify is similar to the initial part of CHANGE mode, producing an .INC file. The editor and translator are not invoked however, and no changes to OBC or SMT occur.

After determining whether the changes made to the .OBC are desired, either ACCEPT or REJECT mode is selected. When ACCEPT mode is selected, the data in the INC file is appended to the specified .DBT file (which is created if needed.) This results in a .DBT file which, along with the .SMT and .TPG, is input to the UPDATE task to produce a .TPG which corresponds to all the changes which have been made to the .OBC, either by Modify/Change or the Untranslator. When the changes are REJECTed, however, the most recent versions of the .OBC and .SMT, which contain the undesired data, are deleted (except that at least one copy is always retained).

CAUTION

> When a translation occurs, either from the Translate page or via Modify, the produced .OBC and .SMT files are interdependent. Be careful not to interfere with this dependency by performing individual operations on one. Modify always performs operations in pairs and is thus the safest course. When Change or Reject Mode is executed, consistency between the two files is checked. If it is not valid, the translation time-stamp is printed and the task does not proceed.

9.3.6 Update

The Update mode is used to incorporate those changes tried during the debug session and/or Modify session which were saved in the .DBT file, and to merge them into the .TPG file. Refer to the description of the Update options page in para 9.5 for details of making the selections.

9.3.7 Debugging the Test Fixture

The fixture can be debugged to a large extent by running the board test program, shorts and opens test, on a bare board (or fully assembled board). The labels required to do this would be found by looking at the .TPG in LEDIT utility mode to find the label of the first opens or shorts test and the label of the test following the shorts test. The shorts test has the label "PRINT." A program that contains only the opens and shorts tests can be generated from the ATG options page. This program will have the extension .CTP. when a board is used, it should not be drilled or, if it is, it should have a sheet of plastic placed over it to allow the vacuum to pull the board down.

Be sure to check the fixture for vacuum leaks around the edge of the board and around the fixture box. This could avoid many intermittent and phantom faults.

A more complete test for fixture wiring would be to generate a fixture test by selecting the "Fixture Test" option on the ATG page. This test includes an opens test that is run with a shorting plate mounted on the fixture and a shorts test that is run with nothing mounted on the fixture. Every nail specified in the .CKT is checked to a common point for both opens and shorts. The test program that ATG generates normally would bear the board name with the extension .FTP.

Fixture debugging can also be enhanced by the use of the probe. The wiring of a particular nail is confirmed by typing the debug command LOCPIN and then touching the desired nail with the probe. The system will then display the receiver test pin to which the nail is wired. A thorough check would consist of probing every pin on the board. Experience may show that this is a worthwhile procedure.

A quicker, but less thorough, probing procedure would be to use the debug command PROBE = ALL for a debug run. The testing will stop at each IC test and prompt the user to SCRATCHPROBE the IC pins. This method may be adequate for a board that consists mostly of ICs.

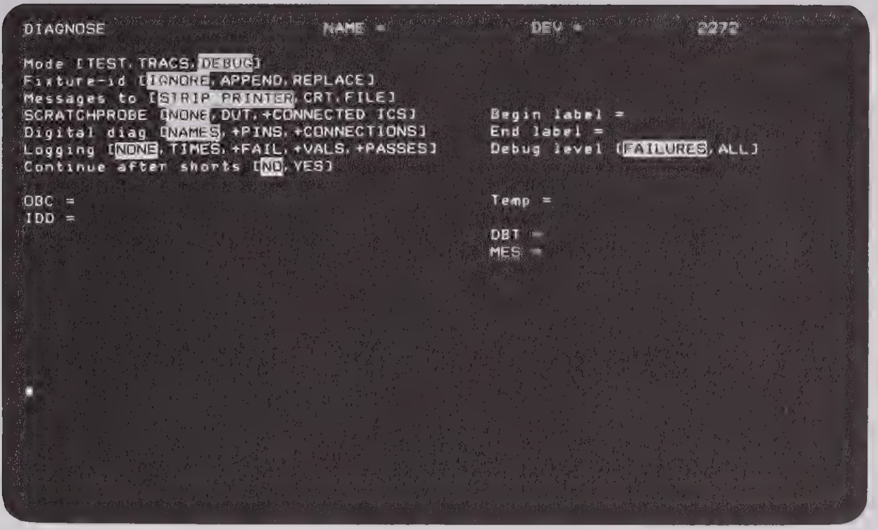
It is advisable to check a fixture for proper power levels before running a test with a loaded board. This could be done by running debug on a bare board with the "Debug level" set to ALL. The testing will stop at each component label and a voltmeter can be used to check voltages on the board.

9.4 DIAGNOSE MODE/UNLIMITED ACCESS

9.4.1 General

The Diagnose options page (Figure 9-7) is used to run a test program in one of 3 modes: testing a pc board, testing a pc board with TRACS, or debugging a test program. DEBUG can be selected from the unlimited access level Diagnose page. The limited access Diagnose page does not have the mode selection option. Refer to Section 2 for a description of the monitor access levels.

It is assumed that the options that are used for testing are already familiar to the user. So, only the options added to the Diagnose page when DEBUG is selected are defined here. Refer to the System Operation manual for test option definitions.



```
DIAGNOSE                    NAME =             DEV =         2272

Mode [TEST,TRACS,DEBUG]
Fixture-id [IGNORE,APPEND,REPLACE]
Messages to [STRIP PRINTER,CRT,FILE]
SCRATCHPROBE [NONE,DUT,+CONNECTED ICS]       Begin label =
Digital diag [NAMES,+PINS,+CONNECTIONS]      End label =
Logging [NONE,TIMES,+FAIL,+VALS,+PASSES]     Debug level [FAILURES,ALL]
Continue after shorts [NO,YES]

OBC =                                        Temp =
IDD =
                                             DBT =
                                             MES =
```

Figure 9-7. Diagnose Monitor Task Options Page.

Mode [TEST,TRACS,DEBUG]
This option enables the desired sub-mode, as previously explained.

Begin label =
The beginning and ending labels define the portion of the test program being operated on by the Debug sub-mode. Specifying no beginning label will cause Debug to start at the beginning of the test program. Specifying an * will cause Debug to start at the first label encountered. The beginning label indicates the first statement in that program segment.

End label =
Specifies the name of the label where the Debug operation ends. No entry causes the end of the test program to be used as the ending label. An * will indicate that the first label following the beginning label is to be used as the end of the program segment.

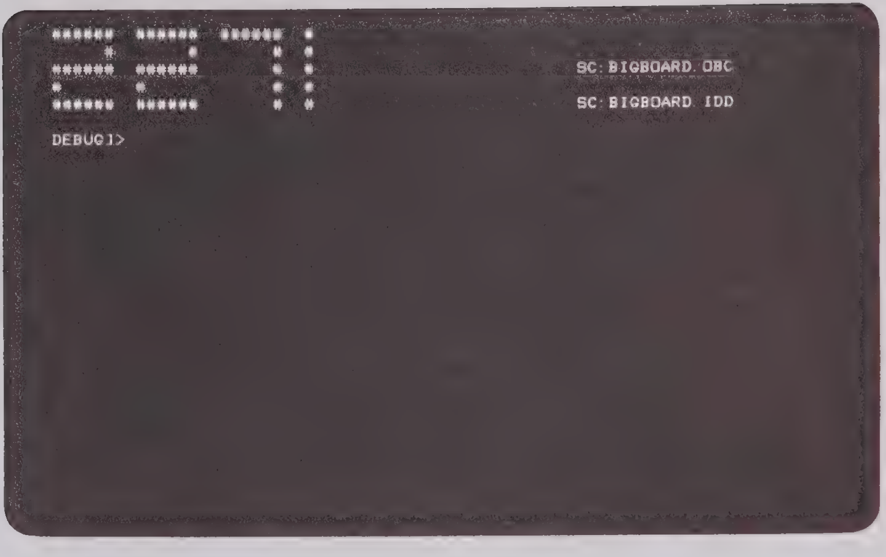
Debug level [FAILURES,ALL]
Selects whether only the test statements that produced errors or all the test statements in the program from Begin label to End label are included in the digital display which appears on the screen following a debug run.

[FAILURES] Specifying FAILURES lists only the test statements in the .TPG which produced errors.

[ALL] Selecting ALL includes all test statements, whether or not they produced or sensed errors during the burst.

DBT=

Once the options page, shown in Figure 9-7, is set up and the RUN key is pressed, the options page will disappear. A message is displayed that asks if a backup copy of the .OBC file is desired. It is usually a good idea to enter Y to this question so that, if anything happens to the .OBC file during the debugging session, the debugging can re-start without having to re-translate the entire



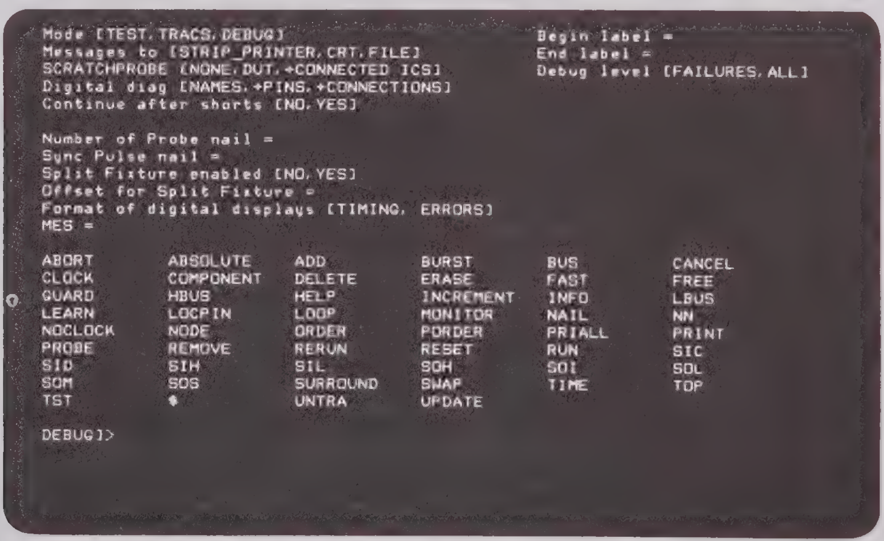
```
******  ******  ******  *
     *       *       *  *
******  ******       *  *                    SC:BIGBOARD.OBC
*       *            *  *
******  ******       *  *                    SC:BIGBOARD.IDD

DEBUG]>
```

Figure 9-8. Lower-Level Debug Monitor Display.

program. After the question is answered the Debug sub-mode lower-level monitor is entered and the display in Figure 9-8 appears.

9.4.2 Debug Sub-mode Commands

Once the Debug sub-mode is active, there are several options that may be specified and commands that may be used. This option list is displayed by pressing the HELP key. The display shown in Figure 9-9 will appear. Many of the options that appear on the Diagnose page are repeated on the Help page. These can be changed for the current debug session but are not permanently changed on the Diagnose page. At the bottom of the Help display is the prompt which indicates that a selection, or a command, may be typed. In most cases, none of the options on this page will need to be entered, so the RUN key can be pressed to execute the test program. The RUN key could also have been pressed



```
Mode [TEST,TRACS,DEBUG]                               Begin label =
Messages to [STRIP_PRINTER,CRT,FILE]                  End label =
SCRATCHPROBE [NONE,DUT,+CONNECTED ICS]                Debug level [FAILURES,ALL]
Digital diag [NAMES,+PINS,+CONNECTIONS]
Continue after shorts [NO,YES]

Number of Probe nail =
Sync Pulse nail =
Split Fixture enabled [NO,YES]
Offset for Split Fixture =
Format of digital displays [TIMING, ERRORS]
MES =

ABORT       ABSOLUTE    ADD         BURST       BUS         CANCEL
CLOCK       COMPONENT   DELETE      ERASE       FAST        FREE
GUARD       HBUS        HELP        INCREMENT   INFO        LBUS
LEARN       LOCPIN      LOOP        MONITOR     NAIL        NN
NOCLOCK     NODE        ORDER       PORDER      PRIALL      PRINT
PROBE       REMOVE      RERUN       RESET       RUN         SIC
SID         SIH         SIL         SOH         SOI         SOL
SOM         SOS         SURROUND    SWAP        TIME        TOP
TST         *           UNTRA       UPDATE

DEBUG]>
```

Figure 9-9. Debug Sub-mode Options and Commands HELP Page.

when the display of Figure 9-8 was present. After execution of the program, the appropriate debug diagnostic data are displayed. The debug prompt (DEBUG]>) will be at the bottom of the display.

At this point there are two general types of commands available: commands to manipulate the diagnostic data for ease of analysis and commands to take action on the test program.

Any of the options on the top-level Diagnose page can be changed from the Debug sub-mode. Changing options from the lower-level monitor must be done using command line directives, as screen-editing works only on the top-level monitor pages. One option is added to the Help page that is not on the Diagnose page. This option is defined as follows:

Format of digital displays [TIMING,ERRORS]

[TIMING]	Displays timing diagrams showing the states of all component pins for each step in the burst. Refer to example in para 9.3.3.
[ERRORS]	Displays a list of logic states of those pins that were sensed to be in error. This provides a more convenient means of listing results on the strip printer. The format of this display is shown in Figure 9-10.

A list of commands that can be used from the Debug sub-mode are listed near the bottom of the page. These commands, which are entered at the DEBUG]> prompt, are described in Table 9-1.

```
Board : DK2:74LS112.OBC          IC : U1    Type: 74LS112

Statement : 7
    5 S 0-1    6 S 1-0

Statement : 8
    5 S 1-0    6 S 0-1

Statement : 29
    7 S 1-0    9 S 0-1

Statement : 30
    7 S 0-1    9 S 1-0
End of Burst
```

Figure 9-10. Display of Logic States.

Table 9-1
DEBUG COMMANDS

Command	Function
ABORT	Stops execution of the test program, resets all test hardware and returns control to the top-level monitor (same as MONITOR command).
ABSOLUTE	Changes the untranslated display of the digital test to show the states of all the nails at each test step, instead of the normal mode of showing the incremental changes of the states of the nails in each test step. This mode is turned off by the INCREMENT command. For example, the normal, or incremental,

Table 9-1 (Continued)

Command — Function

ABSOLUTE (cont.)

display shown in the example in para 9.3.3 would be as follows when the ABSOLUTE command is used:

```
1: IC(6,11) IL(6,11) OS(7) OH(7);
2: IL(6) IH(11) OH(7);
3: IH(6,11) OL(7);
4: IH(6) IL(11) OL(7);
```

ADD stmnt

Adds a new digital statement to the current BURST before the specified statement. The following command, for example, inserts the statement before the current statement 12.

```
DEBUG]> ADD 12 IC(291,353,365) IL(291) IH(353,365);
```

If the statement number was omitted from the command, the system will ask for it when the command is submitted. Special syntax exists to ADD specific nails to a group, without having to retype the nail list. For example, to add a nail to the nail-list of statement 12,

```
12: TEST SHORTS AT (ANA=1-15,22-44,63)
```

all that is required is the command

```
DEBUG]> 12 ANA=+16
```

Commands with no parameters, such as FORCE, are added by specifying just the keyword, for example

```
DEBUG]> FORCE
```

The first time this command is entered, FORCE is added to the statement. Entering the command again removes FORCE from the statement.

To add a PU (or PD, PB, PN) to a statement, the syntax is

```
DEBUG]> PU
```

ADD SET CLOCK

Add a clock statement after the begin label of the current test. The statement inserted is the last clock statement executed by the program. Displays this clock statement.

BURST [=ntimes]

Continually executes the current digital BURST to allow the user to watch the test on an oscilloscope. If BURST=ntimes is entered, there is a delay at the end of execution of the BURST, to allow the hardware and the UUT to cool before the next BURST. This delay is equal to 2(ntimes) x ITT. A number followed by a colon may preced BURST to set up a trigger to sync the scope. The number is the step where the trigger occurs.

Table 9-1 (Continued)

Command	Function
BUS [=nail]	Executes the current BUSTEST, including all the FORCE sections of the test, even if the GO/NOGO part of the BUSTEST did not fail. If a nail number is specified, only the BUSTESTs and the FORCEs for that nail will be executed.
CANCEL	Removes the logging results of the last execution of the test from being recorded in the LOG file, if Logging is enabled.
CHa=n	Sets the scanner, n, to the MUX channel, a.
CLOCK	Displays the current clock timing diagram.
COMPONENT =cname COM =cname	Prints to the MES file all interconnect data for the component specified by name, according to the .IDD file. To display only the data about one pin of a component, type COM=cname.pin.
DELETE stmnt	Deletes the specified statement from the current test. Any displayed statement (either Analog or Digital) may be deleted. A special syntax exists for deleting a nail from a nail list. For example, to delete a nail from the statement

```
12: TEST SHORTS AT (ANA=1-16,22-44,63)
```

the required command to perform the deletion is

```
DEBUG]> 12 ANA=-22
```

Commands with no parameters, such as FORCE, can be deleted from a statement by specifying just the keyword. For example,

```
DEBUG]> FORCE
```

when entered the first time deletes FORCE from the statement. Entering the command again adds FORCE to the statement.

Command	Function
ERASE	Clears the video display except for the DEBUG]> prompt.
FAST stmnt	Inserts the statements 'FAST' and 'END FAST' after indicated statement. Thus a fast section is set up so that fast instructions can be added to the burst with the 'ADD' command. When deleting a fast section, the 'FAST' and 'END FAST' statements cannot be deleted until all other FAST instructions have been deleted.
FREE	The FREE nail command, available from the lower-level monitor, lists up to 5 unused nail numbers that may be added to an existing test set. The command is designed to help a user make test set modifications, when necessary. The system responds to a FREE command by displaying either one of 2 unused nail lists, depending on when the command was issued in the debugging session. At the start of the debugging session, when the system is in the LOADED/RESET state, a GLOBAL

Table 9-1 (Continued)

<table>
<tr><th>Command</th><th>Function</th></tr>
<tr><td>FREE
(cont.)</td><td>
<p>free nail list is generated, listing all unused nails that do not conflict with any test nails throughout the test program. During the debugging phase, however, when debugging a test procedure, a LOCAL free nail list is generated listing all unused nails that do not conflict with nails used in the present test procedure. Note the more stringent restriction on the GLOBAL list (i.e., no conflict with any test) compared to the lesser restriction on the LOCAL list (i.e., no conflict with one particular test).</p>
<p>All free nails lists are labeled either GLOBAL or LOCAL, depending on the type. Since both lists derive unused nail information from the .IDD file, the accuracy of the lists depends on the accuracy of .IDD. For example, if the user modifies the test program without changing the .IDD file, the free nail list will not reflect those modifications.</p>
<p>The user can issue the FREE nail command from the lower-level monitor by typing:</p>
<pre>   FREE opt-parameter</pre>
<p>The optional parameter is used to specify the number of the first nail that the user wishes to have appear in the list. If left unspecified, the list begins with the first available nail. For example, if all 48 nails in the first 3 16-pin nail groups of a 2272 system are unused, the command</p>
<pre>   DEBUG]> FREE</pre>
<p>will display nails 1, 2, 17, 18, 33. Whereas, the command</p>
<pre>   DEBUG]> FREE 3</pre>
<p>will display nails 17, 18, 33, 34, 49.</p>
<p>If the user has recently connected new nails, but has not yet updated the .IDD file to reflect these new connections, this feature could be used to skip over those nails.</p>
</td></tr>
<tr><td>GPRI</td><td>Same as PRIALL</td></tr>
<tr><td>GUARD =nlist</td><td>Command which executes the commands CHC=nlist, CHD=nlist. (For 4-terminal measurements, CHC is the GUARD SENSE channel and CHD is the GUARD channel for the standard MUX connections from ATG generated programs).</td></tr>
<tr><td>HBUS [=nail]</td><td>Executes the current BUSTEST, including all the HFORCE sections of the test, even if the GO/NOGO part of the BUSTEST did not fail. If a nail number is specified, only the BUSTESTs and the</td></tr>
</table>

Table 9-1 (Continued)

Command	Function
HBUS [=nail] (cont.)	HFORCEs for that nail will be executed. Displayed results include all current measurements that would typically be made as part of the BUSBUST diagnostic feature. Refer to Test Programming manual for a description of bus testing.
HELP	Displays the help pages.
INCREMENT	The opposite of ABSOLUTE, this command causes digital nail lists to be printed as changes from the previous step.
INFO [=file]	Allows user to write any test to the printer or a file. This can be used to write notes to a file during a debug session or to annotate diagnostic reports (repair tickets) during a testing run.
LBUS [=nail]	Executes the current BUSTEST, including all the LFORCE sections of the test, even if the GO/NOGO part of the BUSTEST did not fail. If a nail number is specified, only the BUSTESTs and the LFORCEs for that nail will be executed. Displayed results include all current measurements that would typically be made as part of the BUSBUST diagnostic feature. Refer to Test Programming manual for a description of bus testing.
LEARN	Learns the states of all OU nails of the current BURST, or the states of all OS nails if the current BURST does not contain any OU nails, and merges the sensed states into the current test. This command is typically used to generate tests for ROMs. If it is desired to re-run LEARN with a replacement IC, be sure to re-run the burst with the new IC before re-running LEARN.
LOCPIN	Tells the user what nail(s) he is touching with the probe, to enable easy debug of fixture problems. This command can only be executed from the "RESET" or "Waiting for Start" state.
LOOP	Loops the program between the Begin and End labels, executing all instructions between the labels.
MONITOR	Stops execution of the test program and returns the user to the 227x monitor (same as pressing the MONITOR ENABLE key or the command ABORT).
NAIL =	Gives the user connection information about the specified nail. The user enters the number of the test nail in question, after the test has been executed but before the next test has begun, for example,

```
DEBUG]> NAIL 53
```

Table 9-1 (Continued)

Command	Function
NAIL = (cont.)	The system responds by displaying the following information: 1. The circuit node to which that node is connected (e.g., N132 from the .CKT file); 2. Component interconnections to that node (if an .IDD file is available); 3. The receiver connector block and pin to which that nail is connected (e.g., 7-049 means pin 49 on connector block 7); 4. The scanned driver/sensor channel (A, B, C or D) driving that nail.
NN (or NOS)	Causes the test program to start execution without the system attempting to apply the vacuum to the test fixture.
NOCLOCK	Display clock statement without the clock timing diagram.
NODE nodename	Displays the test nail connected to the specified node, and the information described in the NAIL = command.
NU	Probe nail number is displayed. Not listed on HELP page.

ORDER =nlist	Causes the digital IC timing displays to be presented by fixture nail number, in the order listed in the command. For example, typing the command
	`DEBUG]> ORDER = 1,2,5,6,3,4,7,8`
	would show the timing diagram in nail order
	1 2 5 6 3 4 7 8.
	Typing the command with an asterisk
	`DEBUG]> ORDER = *1, 8-10`
	would show the timing diagram for nails 8 through 10 immediately after nail 1 and all others in the current order.
	The command
	`DEBUG]> ORDER = *1, 8-10; *16, 11-12`
	is an example of typing two or more commands in one instruction.

PORDER =plist	Similar to the ORDER command, except the ordering is by component pin number rather than by nail number.

Table 9-1 (Continued)

Command	Function
PRINT [=file]	Prints or copies to a file, the currently displayed timing diagram or other type of debug display.
PRIALL [=file]	Same as the PRINT command except all pages of debug data from the current debug session are printed or copied to a file.
PROBE =cname	Probe the component specified by cname using the Scratchprobing technique. If cname is "ALL", all digital components will be probed. This command can only be executed from the "RESET" or "Waiting to Start" state. It is used to verify system wiring.
REMOVE	Restores the current test to what it was prior to the current set of changes that have been made to this test. REMOVE only has a memory of one active component, which means that the original state of the current test is saved and can be restored at any time, until changes are made to another component test. REMOVE=CLEAR clears the REMOVE buffer to insure that the current test cannot be accidently restored to an unwanted state. REMOVE=INFO tells what test is currently stored.
RERUN [=ntimes]	Causes the current test to be executed again. Pressing the RETURN key has the same effect. RERUN=n executes the current test ntimes times, printing pass/fail statistics, and measurement statistics if they are available. The rerun command causes the debugger to go back to the previous begin burst label. If the test has more than one burst the rerun will not go back to the first burst, so the whole program would have to be restarted.
RESET	Causes the test program to halt, and returns the system to the "RESET" or "Waiting for Start" state. Pressing the RESET key has the same effect. Unlike the MONITOR and ABORT commands, this command will not return you to the 227x monitor.
RUN	Begins execution of the test program, or continues with the execution of the test program.
SIC, SID, SIH, SIL, SOS, SOI SOH, SOL, SOM	Commands to insert single step changes into a burst. These are not test language commands and cannot be inserted in a test program (.TPG) file. However, when entered from the DEBUG]> prompt (after using UNTRA), they function like the usual statements IC, ID, OS, OI, etc, except that the state immediately prior to the command is restored immediately thereafter. This means that, when UPDATE is used with single step commands, the test language command sequence that results in the test program would have to reflect the changes in state. For example: if the state context of component pin 3 is low (IL(3)) and the DEBUG command SIH(3) is inserted, the statements that appear in the updated .TPG are IH(3) in the modified burst step and IL(3) in the next step.

Table 9-1 (Continued)

Command	Function
STEP	Executes the next test language instruction, stopping and displaying the results of the test statement until the next command is entered. All statements in a digital burst, up to the END BURST; statement are executed by the STEP command. Not listed on HELP page.
SURROUND =nail	Prints the names of all the components connected to the specified nail and lists the nails connected to the other pins of the components. For example, if nail 5 connects to U31 pin 1 and U12 pin 2, nail 2 connects to U31 pin 2, etc, the command `DEBUG]> SUR = 5` prints out the following message
SWAP	Swaps the nail lists for CHA and CHB in the current scanner statement, and deletes the nail lists of CHC and CHD.
TIME	Displays system date and time.
TOP	Displays top of debug display.
TST	Changes a digital SET LGC statement in the current statement to a TEST LGC statement.
$	Changes the current TEST LGC statement to a SET LGC ($) statement.
UNTRA	Untranslates the section of the test program which produced the digital timing display, and displays the untranslated test program statements on the screen. If the burst contains more than 32 connect or surce patterns the untranslate cannot be performed and the message TOO MANY CONNECT or SENSE PATTERN will be displayed. A connect pattern exists each time another list of pins is specified by IC and OC statements. A sense pattern exists each time another list of pins is specified by IS and OS statements.
UPDATE	Displays .IDD data and allows for changing that data.

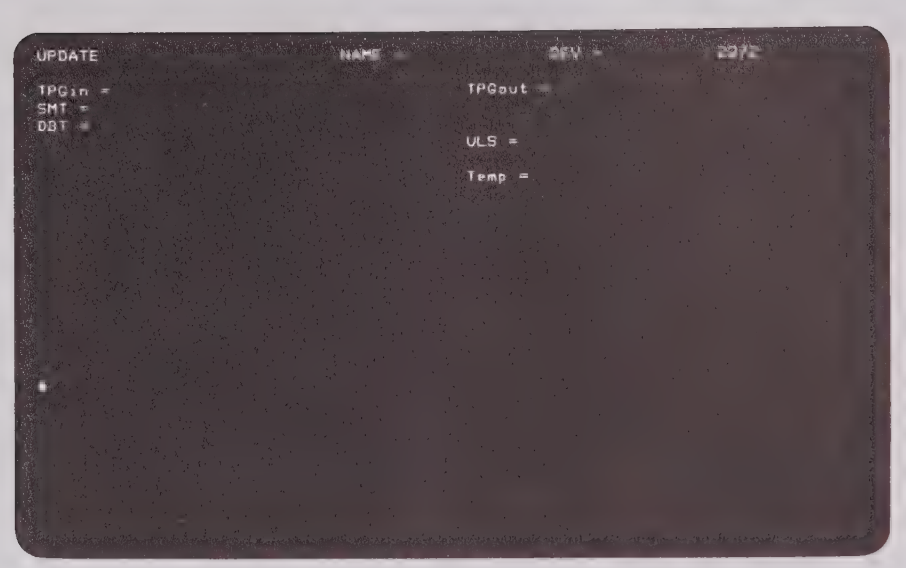
There are a few additional debug commands that are not listed on the HELP page. Execute these commands by pressing the appropriate function key, as described in Table 9-2.

Table 9-2
ADDITIONAL DEBUG COMMANDS

Command	Function
← (left-arrow)	Back - Display the previous page of the digital burst report.
RESTORE Key	Display the burst timing diagram or Shift the digital display one statement left.
→ (right-arrow)	Forward - Display the next page of the digital burst report.
SAVE Key	Forward - Shift the digital display one statement right.
↑	Up - Display previous pin (or nail) group of digital display.
↓	Down - Display next pin (or nail) group of digital display.
NEXT Key	Advance the steps being displayed until next error is shown.
LINE FEED Key	Run the test program.
CTRL/Z	Abort current debug sequence.
DEF	Displays lower-level monitor page defaults.
DS=	Same as NAIL=.
GPRI	Same as PRIALL.
NU	Probe nailnumber is displayed.
RU2	Run test using fixture on right.
SER=	Enter a serial number for the test board.
STA	Display current statement number.
SY	Display sync pulse nail number.
TO	Go to top of debug display.
UP	Update .IDD file.

9.5 UPDATE OPTIONS

The Update mode is used after a debugging session in Diagnose mode where sections of the .OBC file have been changed, to bring the source files up-to-date with the changed binary test program. Update options are shown in Figure 9-12.



```
UPDATE                     NAME =             DEV =          2272

TPGin =                                  TPGout =
SMT =
DBT =
                                         ULS =

                                         Temp =
```

Figure 9-12. Update Mode Options Page.

TPGin =
    Name of the source test program file (.TPG) if it differs from the option name at the top of the page.

SMT =
    Symbol Table file, created by the translator, which keeps a record of test program labels and variable names.

DBT =
    Name of the Debug Trace file (.DBT) which is created during a Debug or Modify session. If no abort takes place, the .DBT is reassigned the extention .DBO at the conclusion of Update.

TPGout =
    Name of the updated test program file. Extension is .TPG if there are no errors in the update process and .TPU if there are errors.

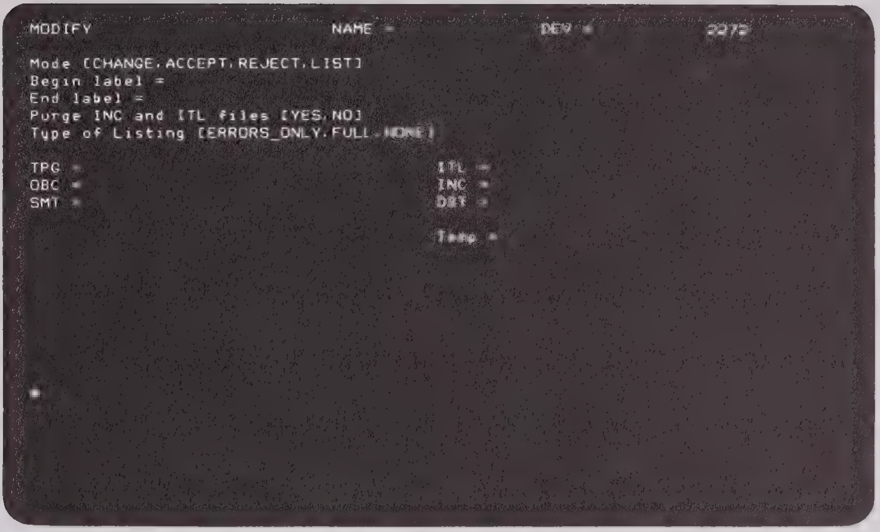
ULS =
    Name of Update Listing file.

Temp =
    Specifies disk device to use for storing temporary files produced by Update.

9.6 MODIFY OPTIONS

The Modify mode allows the user to make changes to a translated test program. A matched set of TPG, OBC and SMT files is required to use the Modify task. The section of the test program between the Begin label and End Label specified by the user is loaded from the TPG file into the Editor.

This section of the test program can then be changed by adding, editing, or deleting individual test statements or entire tests. Leaving the Editor after making changes writes the newly changed program section to the incremental change file (.INC). The translator then processes the INC file to produce an object code version of the changed section. Modify options are shown in Figure 9-13.



```
MODIFY                         NAME =                    DEV =            2272

Mode [CHANGE,ACCEPT,REJECT,LIST]
Begin label =
End label =
Purge INC and ITL files [YES,NO]
Type of Listing [ERRORS_ONLY,FULL,NONE]

TPG =                                              ITL =
OBC =                                              INC =
SMT =                                              DBT =

                                                   Temp =
```

Figure 9-13. Modify Mode Options Page.

Mode [CHANGE,ACCEPT,REJECT,LIST]

[CHANGE] Copies a section of the TPG file into a new or existing .INC file. The .INC file can then be edited. The .INC is automatically translated and appended to the .OBC.

[ACCEPT] When the changes to the test program section are complete and correct, the edited .INC is copied into the .DBT file.

[REJECT] Selecting REJECT indicates the changes made to the program in the .INC file are not acceptable. All versions of the .INC file are deleted and the latest version of the .OBC and .SMT files are deleted.

[LIST] Lists the section of the .TPG file from the Begin label to the End label on the video display if there is no .INC file. Otherwise, the latest version of the .INC file is listed.

Begin label =

Specifies the beginning label of the test program section to be changed.

End label =

Specifies the ending label of the test program section to be changed.

Purge INC and ITL files [YES,NO]

Each time the Modify task is run, a new .INC and .ITL file is created. Purging the excess previous versions of these files helps to keep the disk-device clear of unnecessary files.

[YES] Removes all but the latest (highest numbered) version of these files.

[NO] The files are not purged after the Modify run.

Type of listing [ERRORS_ONLY,FULL,NONE]

The Incremental Translation Listing file (.ITL) receives diagnostic and error messages resulting from the section of the test program loaded into the .INC file. This option selects how much information will be included in the listing.

[ERRORS_ONLY] Only the lines containing errors are listed.

[FULL] All test program lines are listed.

[NONE] Specifies NO listing is to be made.

TPG =
The name of the test program source file. Defaults to the "DEV =" and "NAME =" options for device and name, and .TPG for the extension. An alternate device, name, and/or extension can be specified for the file in this option.

OBC =
The name of the binary test program file. The device and name default to those at the top of the page; the extension defaults to .OBC. An alternate device, name, and/or extension can be included in this option.

SMT =
Symbol table file, created by the translator containing names used in the test program. It is required for subsequent incremental translations and should be maintained consistently with the .OBC.

ITL =
System and error messages produced during a Modify task run are sent to the Incremental Translator Listing file. The file specification defaults to the "DEV =" and "NAME =" entries and the extension to .ITL. The video display (TI:) may be specified, but no permanent (file) copy is made of the messages.

INC =
The section of the .TPG file from Begin label to End label is loaded into the INCremental translation file. The file spec defaults to the device and name at the top of the page and the extension to .INC.

DBT =
The name of the Debug Trace file which is created or appended when entries in the .INC file are accepted by running ACCEPT mode. This file can then be used by UPDATE to correct the .TPG file.

Temp =
Temporary files created by Modify are stored on the device specified in this option. The default is the "DEV =" device at the top of the page.

Production Test Information – Section 10

10.1 GENERAL

Information about the test set must be documented so that production test personnel can find files, fixtures, etc. Special testing procedures, such as filling in Diagnose options, and operator responses may be required. Test data logging or TRACS analysis instructions may also be recommended. There are at least 3 methods of providing this information.:

Use a LOAD subroutine.
Use a batch file.
Provide an instruction sheet.

Examples of these methods are presented in this section.

10.2 LOAD SUBROUTINE

A LOAD subroutine is a part of the test program which is executed only when the test program is loaded into the Run-time system, in other words, when a test run is just beginning. The subroutine is automatically run and can display all operator information and instructions necessary to start the run and can also provide information for operator responses. If the user prepares an UNLOAD subroutine, it is automatically executed when the test run is completed. This routine can provide additional instructions to the operator.

10.3 BATCH FILE TESTING

10.3.1 General

A batch file, also called a batch job, can be created that sets up all options on the Diagnose monitor page and initiates the running of the test program. This reduces the amount of information needed to be conveyed to the operator. It also allows options to be set on options pages that are restricted in LIMITed monitor operation.

A batch file (.BCH) contains a series of options pages which can be run in the order presented in the file. When the batch job is run, the listed options are automatically set. If a run command exists at the end of a page, the applicable task is run automatically. If it does not exist, execution of the job pasuses for the operator to initiate or skip the task. A task can be automatically skipped, if the previous task had a fault, by the insertion of the appropriate command. A listing file (.BLI) is created as the batch job is executing. This .BLI file is a record of all commands executed, resulting error listings, and program messages. The listing file is useful when long batch jobs are run in the user's absence (for instance, overnight), to allow the user to determine the status of the batch job run.

The functions of the keypad are limited to running Diagnose mode and starting batch jobs. Removing the keyboard and enabling the LIMITed access level allow degrees of control over the use of the system. Though the monitor's LIMITed access level forces the user to stay in Diagnose mode, the system manager can allow operators to access any major mode or utility by creating batch files that call the options page and assign option values. Batch mode is not restricted by the LIMITed monitor, that is, any mode can be accessed by a batch job regardless of which access level (LIMITed or UNLIMITed) is enabled.

10.3.2 Running a Batch Job

A batch job can be started from the command mode by typing a period (.) as the first character on the line. Thus, typing

```
#.1731
```

will run the batch input file SC:1731.BCH (assuming the "DEV =" option at the top of the page is SC:). Note that periods and numerals can be typed from the keypad, so numerically named batch files on the default device can be run from the keypad as well as from the keyboard.

A batch job with the same name as that in the "NAME =" option may be started by entering only a period with no name. If the name at the top of the page is SAMPLE and the device is SC:, entering

```
#.
```

directs the system to look for and run the batch job SC:SAMPLE.BCH.

A batch job with the same name as the fixture-id may be started by entering two periods with no name. Assuming the device at the top of the page is SC:, entering

```
#..
```

directs the system to look for and run the batch job SC:fixture-id.BCH.

NOTE

If no fixture-id is wired into the test fixture, an attempt to read it returns the value 0 (zero). Thus, a batch job named 0.BCH can be started by this method even if no fixture-id has been wired into the test fixture.

A feature of batch processing is the automatic running of a batch file named STARTUP.BCH on the system default device (usually SY:). Every time the system is initialized it will look on the default device for a STARTUP.BCH file. If one is found it will be run immediately after the Diagnose mode page appears on the screen. This provides a method of displaying any monitor page (not just Diagnose) when the system is initialized. For example, if the STARTUP.BCH file contained the command

```
#SETUP
```

the page displayed on the screen after initialization is the Setup utility page.

10.3.3 Creating a Batch Job

The monitor save commands, can be used to create a batch input file, which consists of commands that would normally be typed on the individual options pages. Any mode controlled by the monitor may be run under Batch by including the mode name, mode options, and the command RUN in a batch input file.

The procedure for creating the batch input file, using the monitor save command, is:

a. Obtain the appropriate options page.

b. Select the appropriate options on the page.

c. Type the command SAVE or SAVER. When SAVE is used, the options page is saved but is not automatically run while the batch file is executing. The batch processor will display the saved options page, but will not run the mode until the operator types a RUN command. When SAVER is used, the options page is automatically run without waiting for an operator response.

d. Repeat steps a through c for each options page that is to be part of the batch file.

e. Combine the batch files with the Copy utility. Omit this step if the .BCH file is created manually.

When the Batch utility is being run, statements in the input file are executed sequentially, as if they were being typed from the keyboard. Comments may be used but must be bracketed by the symbols /* and */. They may appear alone on a line or after a command and are ignored by the batch processor. Batch input files created with the monitor save commands contain all the options which can occur on the options page, but options which are not displayed on the page are commented out. This allows the user to change the batch input file (.BCH) easily by using the Editor to remove or insert comment symbols. The following example is the result of creating an input file with the SAVER command.

```
#DIAGNOSE
#NAME =SAMPLE
#DEV =SC:
#Mode=TEST
#Fixture-id=IGNORE
#Messages=STRIP_PRINTER
#SCRATCHPROBE=NONE
#Digital=NAMES
#Logging=NONE
#Continue=NO
/*#Maximum=*/
/*#Board=*/
/*#Process=*/
/*#Suppress=NO*/
/*#Analysis=NO*/
/*#Begin=*/
/*#End=*/
/*#Debug=FAILURES*/
#OBC=
#IDD=
#Temp=
/*#LOG=*/
/*#DBT=*/
/*#MES=*/
/*#Qtemp=*/
#RUN
```

The Editor can also be used to create a batch file. Be sure to include the prompt # before each option and command.

RUN commands preceded by a dollar sign ($), instead of #, are executed only if the last RUN command was successful; otherwise, these command lines are skipped. In this context, a job is the execution of an action command (e.g., ATG RUN) for a given set of options. This feature might be used, for example, to perform several ATG runs in sequence and then to enter the Translate mode only if the previous ATG run did not fail.

Once the command SAVER or SAVE is typed after the command prompt (#) at the bottom of the page, all the options on the page are saved in a file whose name and device are that shown on the top of the options page and whose extension is .BCH. When all applicable options pages have been saved, there is a file for each page. These files may have different names, depending on the name that existed at the top of the options page when it was saved. Some files may have the same name with different version numbers (when a page is saved and there is already a .BCH file with the same name, a new file is created with the next higher version number.) All the files created must be combined into a single batch file. This is done by using the Copy utility.

The method just described presents a problem in keeping track of the file names of each saved options page. A more convenient method may be to use the SAVER command with a filespec. The SAVER commands could then provide a series of simple, easy-to-remember filenames such as A, B, C, etc. These save commands would be similar to

```
#SAVER SY0:A
```

and would create a file with the filespec SY0:A.BCH. If 3 files were saved with the names A, B and C, the copy command to combine these files would be

```
#COPY SY0:BOARD.BCH = SY0:A.BCH, SY0:B.BCH, SY0:C.BCH
```

resulting in a batch input file SY0:BOARD.BCH. If a batch file is created manually, only one file with all options is made.

The batch input file is subsequently run by typing

```
#BATCH SY0:BOARD.BCH
```

All options on the 3 saved pages in the batch input file will be run with no further user control. If the user wants to control the running of a page within the batch input file, the SAVE command used should be similar to

```
#SAVE SY0:B
```

This will cause the batch job to pause before running the page that was saved in file B until the user enters the RUN command.

10.4 INSTRUCTION SHEET

If the LOAD and Batch features do not adequately provide the operator interface, it may be desirable to create an instruction sheet to replace or supplement those features. An instruction sheet can be created via the Editor utility and kept as a system file to be called by the operator. An on-line information sheet has limited use because it cannot be directly accessed from the LIMITed monitor; it can, however, be accessed via the LOAD or Batch feature. A hard copy of an instruction sheet may, therefore, be desirable.

Editors – Section 11

11.1 INTRODUCTION

Files are created and edited by using one of the editing programs available in the system software. The principal editor is a powerful screen editing macro written by GenRad for the TECO editing program supplied with the DEC[1] operating system. This editor is referred to as the GR Terminal TECO editor. It is available for use on the GR Terminal (single user terminal or multi-user, foreground terminal) only. There are 3 other editors available in the system software which are available to the background terminal of a multi-user system. These are:

ED2 editor (para 11.8)	A version of the DEC EDT editor with most of the keys redefined to make the command set as similar as possible to the TECO command set. It is not available at the 227X console keyboard. ED2 uses much less central processor time than TECO; for this reason it should be used whenever testing is being performed by a foreground user. ED2 has 2 modes of operation; the screen or character mode, and the line editor mode. The line editor mode can accomplish certain tasks that are difficult or impossible in the character mode. It is available for users who are more comfortable with this type of operation than with a screen editor.
VT10X TECO editor (para 11.7)	A GenRad macro written for the TECO editor that is designed for the VT10X terminal. Should only be used when testing is not in progress in the foreground.
TEX program (para 11.9)	A simplified editor for creating new files. It allows fast text entry with very little computer access time.

All the editors except TEX are accessed by the editor utility of the system monitor. The editor utility is described in para 11.2. Figure 11-1 summarizes the accessing of editors from each terminal. TEX is described in paragraph 11.9.

Since many of the editing commands are identical between editors, the commands are explained according to the keyboard keys used. There are many function keys and control characters on the standard keyboard that perform the same function in all editors. Some examples are the arrow keys, the delete key, ^A and ^K, among others, where the ^ symbol denotes the CTRL key. There are many editor-dependent key combinations on each keyboard that perform functions unique to TECO or to ED2. Figure 11-2 illustrates the commonality of commands. These commands are explained in the following paragraphs:

---

[1] Digital Equipment Corp., Maynard, MA 01754

- 11.3 Directory of all Editing Commands
- 11.4 Control Character and Escape Character Editing
- 11.5 GR Terminal Function Keys
- 11.6 VT10X Function Keypad
- 11.7 DEC TECO Commands
- 11.8 ED2 Editor
- 11.9 TEX Program Text Entering

TEX, ED2, and TECO each have unique advantages and restrictions. The abilities of each are briefly presented and compared in Table 11-1.

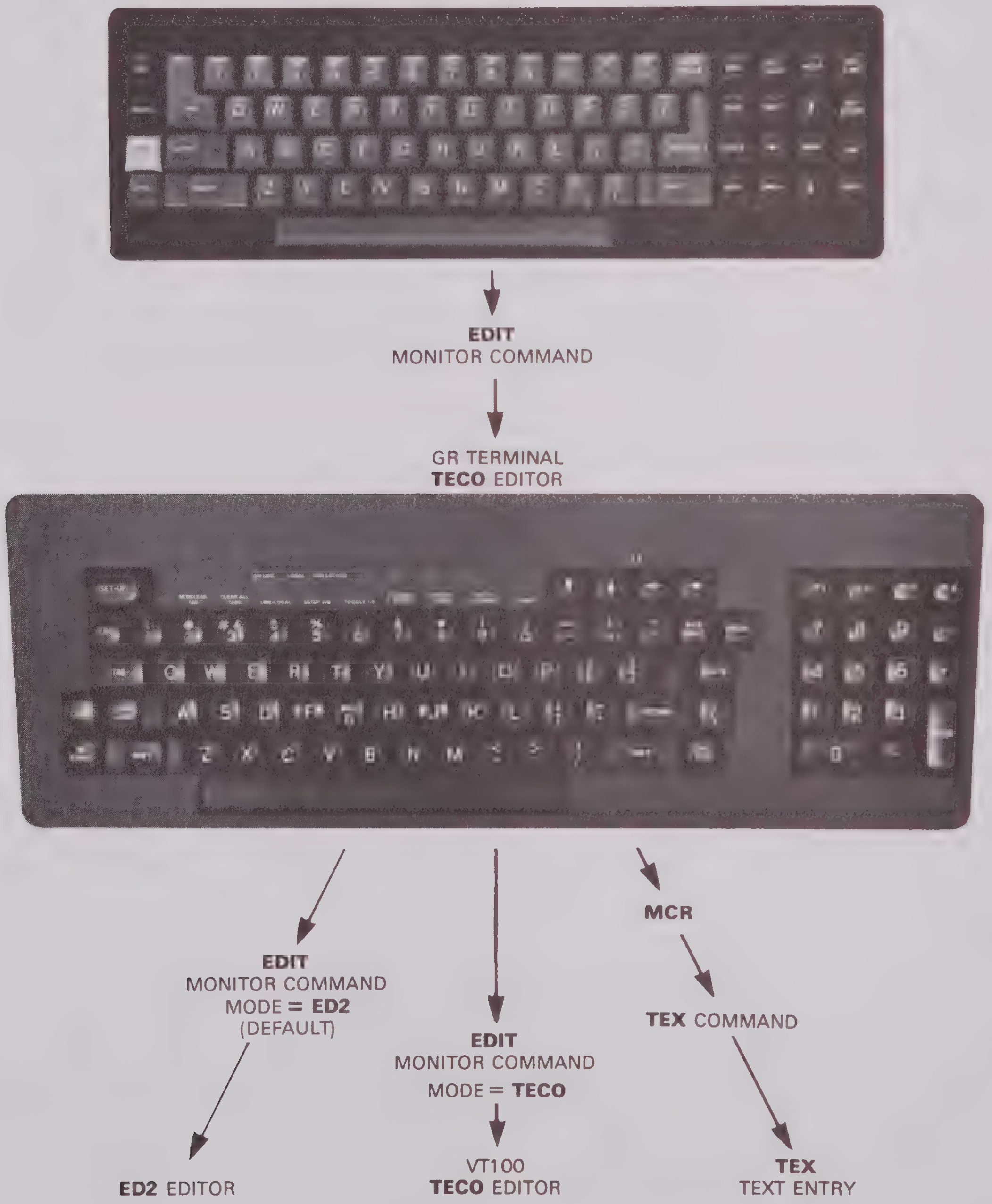

EDIT
MONITOR COMMAND
GR TERMINAL
TECO EDITOR
EDIT
MONITOR COMMAND
MODE = ED2
(DEFAULT)
EDIT
MONITOR COMMAND
MODE = TECO
MCR
TEX COMMAND
ED2 EDITOR
VT100
TECO EDITOR
TEX
TEXT ENTRY

Figure 11-1. 227X System Editor Selections.

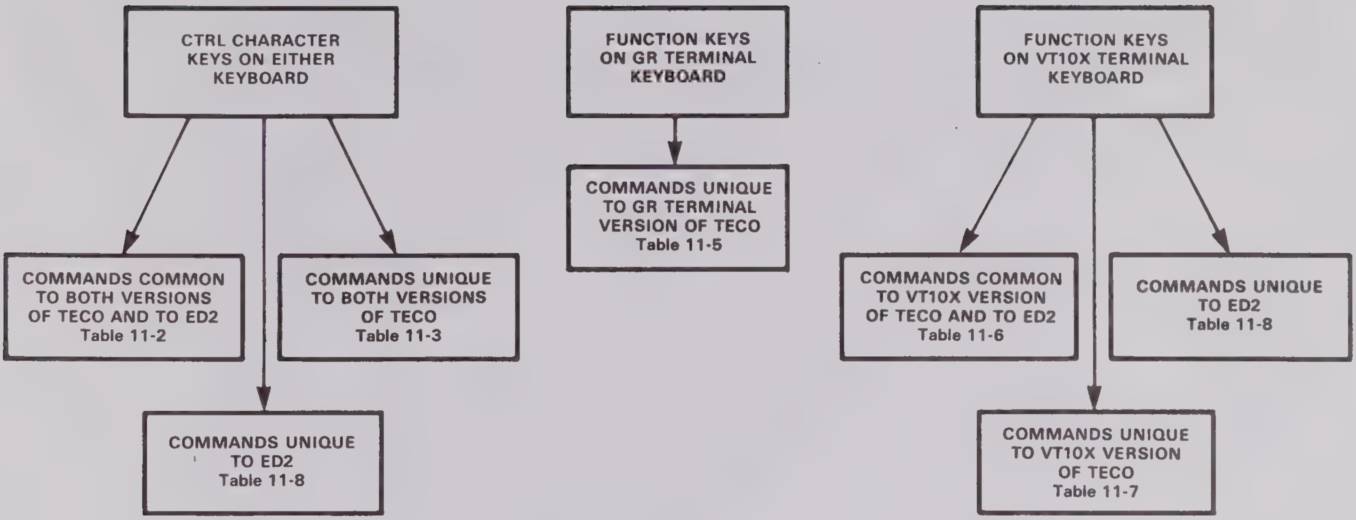

CTRL CHARACTER KEYS ON EITHER KEYBOARD
FUNCTION KEYS ON GR TERMINAL KEYBOARD
FUNCTION KEYS ON VT10X TERMINAL KEYBOARD
COMMANDS UNIQUE TO GR TERMINAL VERSION OF TECO Table 11-5
COMMANDS COMMON TO BOTH VERSIONS OF TECO AND TO ED2 Table 11-2
COMMANDS UNIQUE TO BOTH VERSIONS OF TECO Table 11-3
COMMANDS COMMON TO VT10X VERSION OF TECO AND TO ED2 Table 11-6
COMMANDS UNIQUE TO ED2 Table 11-8
COMMANDS UNIQUE TO ED2 Table 11-8
COMMANDS UNIQUE TO VT10X VERSION OF TECO Table 11-7

Figure 11-2. Commonality of Editing Commands by Keyboard.

Table 11-1
COMPARISON OF FEATURES

	TEX	ED2	TECO
Full-Function Editor	NO	YES	YES
CPU Load	VERY LOW	LOW	HIGH
Screen Mode (Keypad) Editing	NO	YES	YES
Alternate Mode Editing	NO	SOME	YES
User-defined Macros	N/A	SOME	YES
Text Buffer	LINE	FILE	LIMITED LENGTH
Input Multiple Files	N/A	YES	YES

There are several terms that should be understood with relation to editing. Some of these are as follows:

Term	Definition
Buffer	The memory area used by the editor to hold text for editing. When editing an existing file, text is read from the file into the buffer. Text in the buffer may be edited, and when editing is complete, the buffer is written to the output file.
CPU load	The extent that an editor in a background mode impacts and is impacted by currently-running foreground operations.
Character mode	The screen-editing mode of the ED2 editor.
Control character	A special character formed by pressing the ^ key and another key simultaneously. In the editor descriptions, control characters are represented by the symbol ^ followed by the other key to be pressed. For example, the combination of the ^ key and the letter T key is shown as ^T. Standard control characters are defined to perform the editing functions listed in Table 11-2.

Term	Definition
Escape character	A special character formed by pressing the ESC key followed by another key or keys (refer to the definition of n, below). Escape characters are used to perform editing functions in TECO, for example, to enter a number when a command is to be repeated a number of times. In ED2, the 'gold' key (PF1) is used instead of ESC to enter numeric arguments. The ESC key is echoed as a "$" in some modes of TECO.
Full-function editor	An editor with a large enough command set to perform all common editing functions.
Line mode	Applies to the ED2 editor. The line mode is one of 2 ED2 modes that is more suitable for moving large parts of text from within 1, or more files, than is the character mode, but is otherwise not as convenient to use.
Macro	A sequence of commands, assigned a name and stored in a buffer or register, which can be executed using a single command that calls the macro by name.
n	The letter n is used in many editing commands to indicate a number that is taken as an argument in an escape sequence. For example, ESC n ^F is a TECO command to advance the cursor n characters. The command ESC 15 ^F, where n=15, advances the cursor 15 characters.
Page	The text which exists between the start of the buffer or a formfeed character and the next formfeed character or the end of the buffer. If there are no formfeed characters the entire buffer is a page.
Screen editor	An editor which displays a page of text of the screen and allows the cursor to be moved directly to any position on the screen to add, replace, or delete text.

11.2 EDITOR UTILITY DESCRIPTION

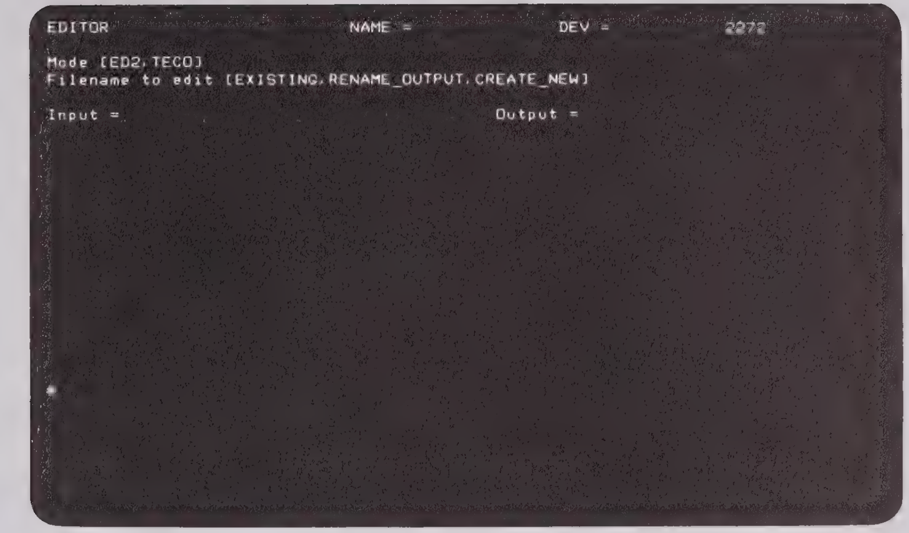
This description contains instructions for selecting an existing file to be edited, for editing a file and giving it a new name, and for creating a file. The reader should be familiar with the basic operation of the monitor. If any of the following items are not understood, refer to Section 2 for instructions on how to use the monitor: Limited/unlimited access; NAME=/DEV=; one line command; option page screen editing.

11.2.1 Accessing the Editor Mode

The Editor mode is called from the UNLIMITed access level of the monitor, using any of the methods described in Section 2. For example, entering

```
#ED
```

causes the Editor options page in Figure 11-3 to be displayed.



```
EDITOR                     NAME =              DEV =          2272

Mode [ED2,TECO]
Filename to edit [EXISTING,RENAME_OUTPUT,CREATE_NEW]

Input =                                    Output =
```

Figure 11-3. Editor Utility Options Page.

Mode [ED2,TECO]

When the Editor is called from the GenRad keyboard, this option is not present and the GR Terminal TECO editor is automatically enabled. If the Editor page is called from the background VT10X terminal, this option appears to allow the user to select either of two editors, ED2 or TECO.

[ED2] Enables the ED2 editor.

[TECO] Enables the VT10XTECO editor.

Filename to edit [EXISTING,RENAME_OUTPUT,CREATE_NEW]

[EXISTING] The default selection, allows an existing file to be opened for editing. The output file created will have the same device, name and extension as the input file, with the version number incremented by 1.

[RENAME_OUTPUT] To rename the output file, enable this selection and the "Output =" file spec option appears to allow the new name to be entered.

[CREATE_NEW] Enable this selection when creating a new file. A blank page will be displayed when the editor is first entered, and the resulting output file will be assigned the version number 1 (assuming no file with the same name already exists).

Input =

The required entry to this option is the extension of the file to edit. The "DEV =" and "NAME =" entries from the top of the page will be used as the device and name of the file if none is specified.

Output =

This file spec option only appears when the "Filename to edit" option is set to RENAME_OUTPUT. An alternate device, file name, and/or extension can be assigned to the output file with this option.

Version numbers are not normally included when input and output files are specified. An output file is automatically assigned the next higher version number by the software if a version number is not specified.

NOTE

If the same version number is assigned to both input and output files, for example,

```
IN=SAMPLE.TPG;5
OUT=SAMPLE.TPG;5
```

then on a normal exit, the input file will be destroyed and replaced by the new edited file. However, if the edit operation is aborted (e.g., ^C, ESC E or disk full condition), then both the old (input) and the new (output) files will be lost.

The RUN command automatically loads the input file into the editor text buffer for editing. If the specified input file does not exist, the editor will start with an empty text buffer, so that a new file can be created.

To use the editor, enter the "Input =" file name and device (and "Output =" file specification, if the output file is to be renamed) and then issue a RUN command. For example,

```
#ED LB1:NAME1.TPG>NAME2.TPG/FILE=RENAME
```

will load the ASCII source file NAME1.TPG from LB1: for editing. After the editing session has been completed, the file will be output to SC: as NAME2.TPG.

Text Editing

Once the RUN command has been issued from the Editor mode options page, the editor is entered, and the file listed as the input file is loaded into the text buffer and the first 24 lines of text (22 lines in ED2) are displayed. If a new file is to be created, the text buffer and display are cleared so that new text can be entered.

11.3 DIRECTORY OF EDITING COMMANDS, BY FUNCTION, FOR ALL EDITORS

Editing consists of several basic functions such as moving the cursor and deleting text. There are several variations of each basic function. There are also variations between editors as to the commands used to perform the functions. This para lists all the commands, by basic function for all editors. Each item in the list references the Tables in which the command is more completely defined.

Both editors supply considerable on-line help to make learning their commands easier. For TECO, press the keys ESC H for general help, or ESC G for definitions of the control character commands. For ED2, press PF4 for general help, PF1 H for help on individual commands.

11.3.1 Cursor Positioning

Function	Command	Editor	Table
forward 1 char.[1]	^F	All	11-2
	→	All	11-6,7
back 1 char.[1]	^B	All	11-2
	←	All	11-6,7
beginning of line	^A	All	11-2
beginning of next line[1]	^N	All	11-2
	0 (function key)	VT10X TECO, ED2	11-7
	SHIFT ↓	GR Terminal TECO	11-6
down 1 line[1]	↓	All	11-2
beginning previous line[1]	^H	All	11-2
	4 (function key)	VT10X TECO, ED2	11-7
	SHIFT ↑	GR Terminal TECO	11-6
	BACK SPACE	All	11-6,7
up 1 line[1]	↑	All	11-2
end current line	^E	All	11-2
end nth line	ESC n ^E	either TECO	11-2
back 24 lines	ESC ↑	either TECO	11-5
	PF1 4	ED2	11-7
forward 24 lines	ESC ↓	either TECO	11-5
	PF1 0	ED2	11-7
beginning of buffer	^T	All	11-2
	1 (function key)	VT10X TECO, ED2	11-7
	^SHIFT ↑	GR Terminal TECO	11-6
nth character in buffer	ESC n ^T	either TECO	11-3
end of buffer	^Z	All	11-2
	2 (function key)	VT10X TECO, ED2	11-7
	^SHIFT ↓	GR Terminal TECO	11-6
	CTL Z (function key)	GR Terminal TECO	11-6
forward 1 word[1]	^W	All	11-2
	CTL W (function key)	GR Terminal TECO	11-6
back 1 word[1]	^J	All	11-2
continuous forward	ESC →	either TECO	11-5
continuous backward	ESC ←	either TECO	11-5
stop continuous action	ESC	either TECO	11-5

11.3.2 Insert Text

char. from keyboard	Type desired characters at cursor position	All	
include file	ESC I	either TECO	11-5
	PF1 I	ED2	11-7
blank line above[1]	7 (function key)	VT10X TECO, ED2	11-7
n blank lines[1]	^M	All	11-2

---

[1] Can be preceded by ESC n in TECO, or PF1 n (gold key) in ED2, to perform operation n times.

11.3.3 Delete

Function	Command	Editor	Table
char. at left[1]	DELETE	VT10X TECO, ED2	11-7
	DEL CHAR	GR Terminal TECO	11-6
	ESC-1 ^D	either TECO	11-2
char. at cursor[1]	^D	All	11-2
	5 (function key)	VT10X TECO, ED2	11-7
word at right	ESC ^D	either TECO	11-5
word to left	ESC - 1 ESC ^D	either TECO	11-5
Delete n words			
at right	ESC n ESC ^D	either TECO	11-5
n words to left	ESC n ESC ^D	either TECO	11-5
to end of line[1]	^K	All	11-2
	^K (function key)	GR Terminal TECO	11-6
	DEL LINE	GR Terminal TECO	11-6
	ESC n ^K (function key)	GR Terminal TECO	11-6
	ESC n DEL LINE	GR Terminal TECO	11-6
of line to cursor[1]	^U	All	11-2
	SHIFT DEL LINE	GR Terminal TECO	11-6
mark to cursor	ESC ^K	either TECO	11-5

11.3.4 Search

char. forward in buffer	^P prompt char.	either TECO	11-2
text forward in buffer	SHIFT SRCH prompt text SRCH	GR Terminal TECO	11-6
	ENTER prompt text .	VT10X TECO, ED2	11-7
	^P prompt text .	ED2	11-7
repeat search	SRCH	GR Terminal TECO	11-6
	.	VT10X TECO, ED2	11-7
nth occurrence of text	ESC n SHIFT SRCH...	GR Terminal TECO	11-6
	ESC n ENTER...	either TECO	11-7
	PF1 n^P .	ED2	11-7
nth occurrence of char.	ESC n^P...	either TECO	11-2
find mark	ESC - ESC F	either TECO	11-5

11.3.5 Search and Replace

replace a text string[1]	^V prompt text prompt text ESC ESC	All	11-2
	prompt text prompt text ESC ESC	VT10X TECO, ED2	11-7

11.3.6 Moving Text

Mark start of save	MARK	GR Terminal TECO	11-6
	ESC 9 (function key)	VT10X TECO	11-7
	PF2	ED2	11-7
save then delete	ESC ^T q	either TECO	11-5
	ESC ^T:q	either TECO	11-5
	PF3	ED2	11-7

---

[1] Can be preceded by ESC n in TECO, or PF1 n (gold key) in ED2, to perform operation n times.

Moving Text (continued)

Function	Command	Editor	Table
save a copy	ESC ^X q	either TECO	11-5
	ESC ^X:q	either TECO	11-5
save a line	PF1	VT10X TECO,	11-7
save n lines	ESC n PF1	VT10X TECO,	11-7
insert saved text	ESC ^G q	either TECO	11-5
	PF3	VT10X TECO	11-7
	PF1 PF3	ED2	11-7
move words from previous line	ESC n ESC N	either TECO	11-5
move words from next line	ESC n ESC N	either TECO	11-5

11.3.7 Buffer/Page Control

write to cursor, read next	NEXT	GR Terminal TECO	11-6
write all, read next	8 (function key)	VT10X TECO	11-7
	^Y prompt Y	either TECO	11-3
Find next form-feed	8 (function key)	ED2	11-7
Find last buffer and cursor position	PF1 L	ED2	11-7
Find main buffer and cursor position	PF1 M	ED2	11-7
write to cursor, read nth	ESC n NEXT	GR Terminal TECO	11-6
write n, read n	ESC n ^Y prompt Y	either TECO	11-3
append next buffer	ESC ^A	either TECO	11-5
append next n buffers	ESC n ESC ^A	either TECO	11-5
append rest of file	ESC-1 ESC ^A	either TECO	11-5

11.3.8 File and Edit Control

write, close, exit	ESC X	either TECO	11-5
	ESC ^Z	either TECO	11-5
delete, exit	ESC E	either TECO	11-5
	ESC - ^Z	either TECO	11-5
Write current buffer to file	PF1 W	ED2	11-7
Same as PF1 W except in print format	PF1 P	ED2	11-7
save, return to cursor	ESC 1 ESC F	either TECO	11-5
	ESC TAB	either TECO	11-5
Save, return to begin	ESC F	either TECO	11-5
mark, save, close	ESC 0 ESC F	either TECO	11-5
leave TECO macro	^C	either TECO	11-3
Abort command in progress	^C	All	11-2
Exit from screen to line mode	^G	ED2	11-4
leave TECO macro	^G	either TECO	11-3
Return to screen edit	mi ESC ESC	either TECO	11-3
	c ESC ESC	ED2	11-4

11.3.9 Miscellaneous

Function	Command	Editor	Table
rewrite the screen	^L	all	11-2
restore text deleted	^R	all	11-2
restore text deleted	^X	either TECO	11-4
restart scroll	^Q	ED2	11-4
stop scroll	^S	ED2	11-4
Redefine command key	PF1 D	ED2	11-7
execute macro	ESC ^E q	either TECO	11-5
execute macro w/value	ESC n ^E q	either TECO	11-5
keypad help display	ESC H	either TECO	11-5
scroll forward	ESC ^N	either TECO	11-5
scroll backward	ESC ^H	either TECO	11-5
see all no print char.	ESC ^L	either TECO	11-5
learn edit sequence	ESC L	either TECO	11-5
set left margin	ESC n ESC M	either TECO	11-5

11.4 CONTROL CHARACTER AND ESCAPE CHARACTER EDITING

Control character editing is defined as editing that is done by using commands that consist of only control characters plus an optional Escape character. The control characters use letters from the standard keyboard and the escape characters are numbers from the standard keyboard. The escape character precedes the control character and is used as an argument to indicate the number of times to repeat the command. Most control character commands perform identical functions in all editors. This common command set makes it easy to go from one editor to another. Table 11-2 lists and defines the common control character command set. Tables 11-3 and 11-4 list and define control character commands unique to TECO and ED2, respectively.

Table 11-2
CONTROL CHARACTER COMMANDS COMMON TO TECO AND ED2

Command	Function
^A	Move the cursor to the start of the current line.
^B[1]	Move the cursor back 1 character.
^C	Aborts a command in progress; control remains in the editor.
^D[1]	Delete the character at the cursor.
ESC n ^D	Delete the character or n characters at the cursor. (PF1 instead of ESC for ED2.)
^E[1]	Move the cursor to the end of the current line.
^F[1]	Move the cursor forward 1 character.

---

[1] Can be preceded by ESC n in TECO, or PF1 n (gold key) in ED2, to perform operation n times.

Table 11-2 (Continued)

Command	Function
^H[1]	Put the cursor at the beginning of the preceding line.
^J[1]	Move the cursor backward 1 word.
^K[1]	Delete from the cursor position to the end of the current line.
^L	Rewrite the screen. This command is useful if the display is interrupted by a broadcast from the system manager or when there is a power glitch in the system.
^M[1]	Opens one line by inserting a carriage return and linefeed sequence above the cursor.
^N[1]	Put the cursor at the beginning of the next line.
^R	Restore whatever text was deleted by the previous ^K or ^U command, starting at the current cursor position.
^T	Move the cursor to the beginning of the text buffer.
^U	Delete the text between the start of the line and the cursor.
^V[1]	Search for one text string and replace with another text string. After typing ^V, the prompt Replace search: is displayed. Enter search characters followed by ESC ESC or . (keypad dot) and then the prompt Replace with: is displayed. Enter the new text followed by ESC ESC or . (keypad dot) and the occurrence of the search string will be replaced with the replace string. If a search and replace cannot be done, a failure message is displayed on the screen. Note that this search and replace is valid only within a buffer.
^W[1]	Move the cursor forward 1 word.
^Z	Move the cursor to the end of the text buffer.

---

[1] Can be preceded by ESC n in TECO, or PF1 n (gold key) in ED2, to perform operation n times.

Table 11-3
CONTROL CHARACTER COMMANDS UNIQUE TO TECO

Command	Function
^C	Exit the TECO macro. Control is returned to TECO (* prompt). Return is via mi ESC ESC.
^G	Exit the TECO macro. Control is returned to TECO (* prompt). Return is via mi ESC ESC.
ESC n ^T	Move the cursor to the nth character from the top of the buffer.
^Y	Outputs all text in the buffer, clears the buffer, and reads in (yanks) the next buffer load for edit. When ^Y is entered, the message ok to write this page and read next? occurs. Answering Y (Yes) executes the command. Any other answer aborts the operation.
ESC n ^Y	Repeats the yank command for n pages.

Table 11-4
CONTROL CHARACTER COMMANDS UNIQUE TO ED2

Command	Function
^G	Switches to line mode. The * prompt is displayed. Return is via c ESC ESC
^Q	Continues output to display if previously stopped by ^S.
^S	Stops output of text to display.

Table 11-5
ESCAPE CHARACTER COMMANDS USED IN TECO

Command	Function
ESC ^A	Attempts to append next page from the input file to the present buffer. TECO will normally draw in text from the input file until its buffer is nearly full, in which case n lines are appended (until memory overflow occurs). However, TECO will stop filling the text buffer when it encounters a formfeed character. Files broken into pages by formfeed characters must therefore be edited page by page, unless this command is used to expand the number of pages present in the buffer.
ESC n ESC ^A	Attempts to append n pages from the input file to the present buffer.
ESC -1 ESC ^A	Will append all pages in the file; however, memory overflow will occur if the file is too large.

Table 11-5 (Continued)

Command	Function
ESC ^D	Delete the word at and to the right of the cursor.
ESC n ESC ^D	Delete n words at and to the right of the cursor.
ESC n ESC ^D	Delete the word or n words to the left of the cursor.
ESC E	Quit TECO and escape from the ATG edit-on-errors loop, without creating a file. The previous version of the file is maintained. Prompts first for a "Y" to confirm user's intention.
ESC ^E q	Execute the macro in Q-register q. A basic TECO command string must have previously been entered into the Q-register by the user
ESC n ESC ^ Eq	'Execute' the macro in Q-register q. Use the value of n as an argument to the macro. The cursor is positioned at the top left of the screen during the execution.
ESC F	Find the top of the file. The file is closed out and then reopened. The cursor is positioned at the top of the first page of the file. If there is no current output file, the terminal bell sounds and the command is ignored.
ESC 1 ESC F	Close and then reopen the file. Position the cursor in the same place it was prior to this command. This command protects the editing performed since the start of the session or since the last ESC 1 ESC F (or ESC F) command, if the system crashes.
ESC 0 ESC F	Close the file and mark (remember) the current cursor position. This command is used in conjunction with the following command.
ESC - ESC F	Find the cursor editing position previously marked by an ESC 0 ESC F command.
ESC ^G q	'Get' (insert) the contents of the temporary buffer q into the text buffer ahead of the cursor. Here q can be any letter from A to Z; the GenRad editing macro uses registers I to K; the last file name used is stored in a buffer named "*"; and the last search string used is stored in a buffer named <_>.
ESC H	Redisplay the keypad layout. The functions of the special-purpose keys are displayed. Type any non-special keypad key to redisplay the text buffer.
ESC ^H	Scroll backward. Press any non-special keypad key to halt the scroll. The cursor is positioned at the beginning of the line.

Table 11-5 (Continued)

Command	Function
ESC I	Insert another file at the cursor location in the file being edited. With the cursor at the desired location, enter the ESC I command. A prompt will appear which requests the name of the file to be inserted. Type the file specification followed by ESC ESC. Once the file is inserted, the cursor will be located at the end of the new file. If the amount of text inserted caused an additional text buffer to be used, the original text buffer will be written to the output file and the remaining text copied into the next buffer.
ESC ^K	If a mark is set (refer to the description for the ESC 9 command in Table 11-7), delete the text between the mark and the current cursor position.
ESC L	Allows TECO to learn an editing sequence of both commands and text. After ESC L is typed, editing commands from this table (Table 11-3) and/or text will be stored as they are executed on the screen. A second ESC L ends the sequence to be remembered. The editing operation just recorded can then be requested as often as necessary by typing ESC Z. The command ESC L ESC L cancels the learned sequence.
ESC ^L	Toggle the see-all display mode. An ESC ^L activates this mode and non-printing characters such as carriage returns and linefeeds are visible. A subsequent ESC ^L turns off this mode.
ESC n ESC M	Set the left margin, for the line on which the cursor is located, at column n. When a left margin is set, enough tabs and spaces are entered automatically to position the cursor at this margin whenever a carriage return or an open line (^M or the 7 keypad key) is pressed. Press ESC 0 ESC M to return margin to first column.
ESC n ESC N	Move the next word or n words from the following line to the current line. The cursor must be at the end of a line.
ESC n ESC N	Move the last word or n words from the previous line to the current line. The cursor must be at the beginning of a line.
ESC ^N	Scroll forward. Press any non-special keypad key to halt the scroll. The cursor is positioned at the beginning of the line.
ESC ^T[:]q	'Cut' (copy and delete) the text between the cursor and the mark (set with ESC 9 -- auxiliary keypad) into the temporary buffer (TECO q-register) q. If the colon is present (brackets do not belong to the command syntax), append the text to the storage buffer, leaving previous text ahead of it. The variable "q" can be any letter from A to H or from L to Z.
ESC X	Writes out the rest of the file being edited and returns to the EDIT Mode Options Page or to an Edit-On-Errors ATG run.

Table 11-5 (Continued)

Command	Function
ESC ^Z	Exit TECO, preserving the output.
ESC - ^Z	Quit TECO. The edited file is not saved.
ESC <TAB>	Same as ESC 1 ESC F.
ESC →	Move cursor continuously forward. Type any character to stop.
ESC ←	Move cursor continuously backward. Type any character to stop.
ESC ↓	Move cursor down 24 lines.
ESC ↑	Move cursor up 24 lines.

11.5 GR TERMINAL FUNCTION KEYS

All editing commands are entered via the keyboard and echoed on the video display. A series of keys on the right side of the keyboard allows many text editing jobs to be handled with a minimum of effort. The keys and their functions are shown in Figure 11-4. The complete set of keys available at the keyboard is shown in Figure 11-1. Each key function is described in Table 11-6. Any of the functions marked with a [1] will repeat n number of times if, preceding the command, the ESCape key is pressed and then the number n is entered on the main keyboard.

Table 11-6
FUNCTION KEY COMMANDS FOR GR TERMINAL TECO

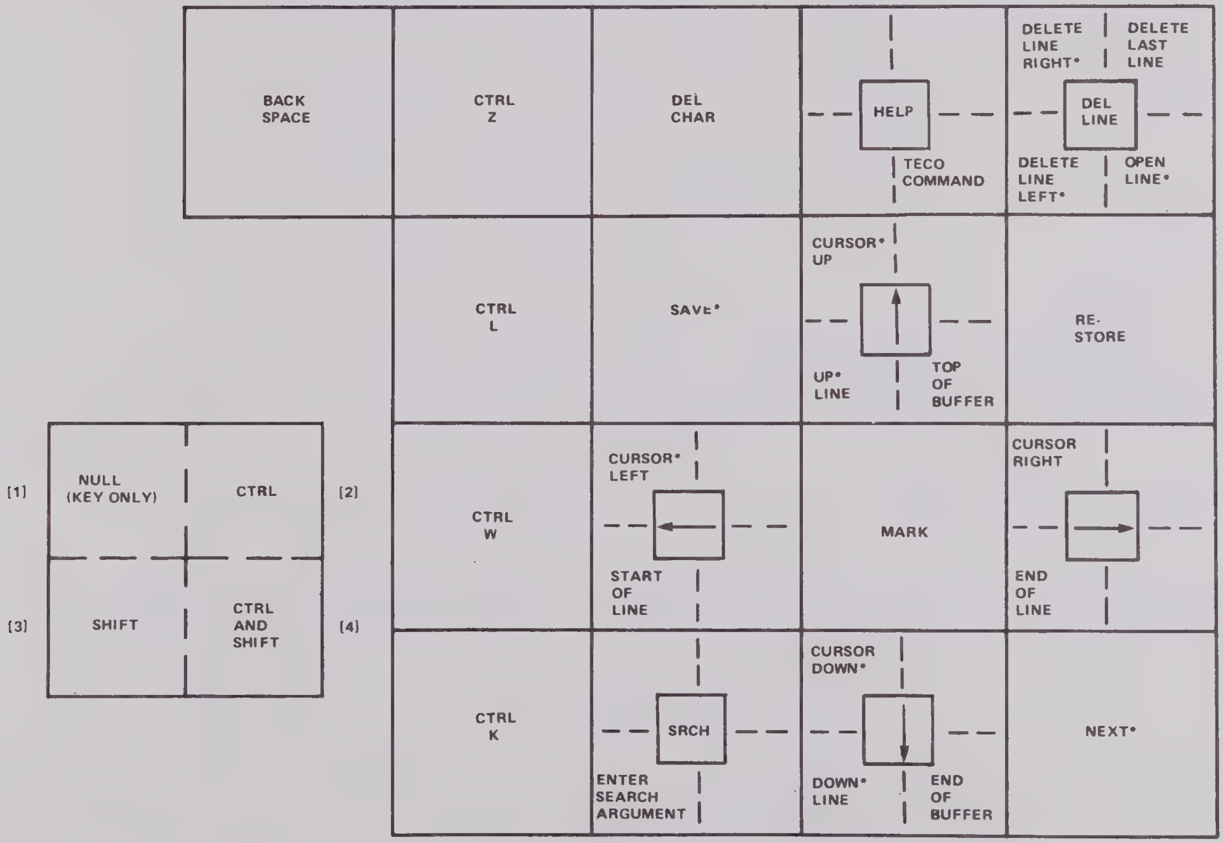
Keys Pressed	Function
BACK SPACE	Moves the cursor up one line (same as ^H).[1]
CTL Z	Moves the cursor to the end of the text buffer.[1]
CTL L	Re-writes the video display screen text.
CTL W	Advances cursor one word, i.e., to the next space, tab, carriage return, or linefeed.[1]
CTL K	Deletes the line of text at and to the right of the cursor.[1]
DEL CHAR	Deletes the character to the left of the cursor.[1]
SAVE	Save a line. Saves text in a register, starting at the cursor position through the next CR-LF. If a mark has been set, saves text between the mark and the cursor position.

Table 11-6 (Continued)

Keys Pressed	Function
ESC n SAVE	Saves n lines of text. The cursor is positioned at the first line or lines to be saved, and then the SAVE key is pressed. The cursor will be re-positioned to the line after the last line to be saved as the command is executed. The text saved is not deleted from the text buffer. This command is ideal for saving and restoring small segments of text (usually 10 lines or less at one time). Note, however, that text entered into the text buffer and not deleted can fill the buffer to a point where a ?MEM MEMORY OVERFLOW * message will be generated. If this should occur, the TECO editor itself will be entered as evidenced by an asterisk (*) display. A user may then enter *MI$$ (where $$ means press the ESC key twice.) to return to the GenRad macro and continue the editing session. Next, the cursor should be positioned just below the text to be output, and a ESC W command issued to output text from the buffer. If the message continues to occur, a user should exit TECO and then re-enter (use ESC and X to return to the GenRad Edit Mode Options Page).
←	Moves cursor one or more characters to the left.[1]
SHIFT ←	Moves the cursor to the start of the line.
SHIFT SRCH	Searches are initiated from the current cursor position. To enter a search command, first hold down the SHIFT key while pressing the SRCH key. A "Buffer search:" message is displayed. At this point, the character string to be searched for is entered and the SRCH key alone is pressed to execute the search. For example, SHIFT SRCH displays the Buffer search: message. A user then enters the character string to be searched for Buffer search: abc and then presses the SRCH key to initiate the search. The cursor is positioned just after the first occurrence of

Table 11-6 (Continued)

Keys Pressed	Function
	"abc" in the text buffer. The character string is "remembered" and pressing SRCH again will result in a search for the next occurrence of the string (or the message "Can't find abc" occurs, and the cursor will not be moved).
SRCH	Executes the search specified from the cursor position and positions the cursor just after the text string located.
ESC - SRCH	Searches backwards toward the top of the buffer.[1]


BACK SPACE
CTRL Z
DEL CHAR
HELP
TECO COMMAND
DELETE LINE RIGHT*
DELETE LAST LINE
DEL LINE
DELETE LINE LEFT*
OPEN LINE*
CTRL L
SAVE*
CURSOR* UP
UP* LINE
TOP OF BUFFER
RE-STORE
[1]
NULL (KEY ONLY)
CTRL
[2]
[3]
SHIFT
CTRL AND SHIFT
[4]
CTRL W
CURSOR* LEFT
START OF LINE
MARK
CURSOR RIGHT
END OF LINE
CTRL K
SRCH
ENTER SEARCH ARGUMENT
CURSOR DOWN*
DOWN* LINE
END OF BUFFER
NEXT*

NOTES

[1] Depress only the keyboard key listed (DEL CHAR, HELP, etc.) to execute.
[2] Hold down the CTRL key while depressing the other key to execute.
[3] Hold down the SHIFT key while depressing the other key to execute.
[4] Hold down the CTRL and SHIFT keys while depressing the other key to execute.
* Depress ESC n keys to move n positions at once.

Figure 11-4. GR Terminal Function Keypad.

Table 11-6 (Continued)

Keys Pressed	Function
^ SHIFT HELP	Allows a Basic TECO command to be entered. The display Command: message. A user then enters the TECO command and executes it by pressing the ESC key twice. ESC is echoed as a dollar sign ($) each time it is pressed. Some TECO commands that may be entered are described in para 11.3.6.
↑	Moves cursor up one line.[1]
SHIFT ↑	Moves cursor to left margin, then up one or more lines.[1]
^ SHIFT ↑	Moves cursor to top of buffer or to mark, if one is set.
MARK	Sets a mark for use with other commands such as ^ SHIFT ^ or SAVE.
↓	Moves cursor down one line.[1]
SHIFT ↓	Moves cursor to left margin, then down one or more lines.[1]
^ SHIFT ↓	Moves cursor to end of text buffer (or to mark, if one is set).
DEL LINE	Deletes all of the line to the right of the cursor.[1] (ESCn deletes following lines entirely.)
SHIFT DEL LINE	Deletes all of the line to the left of the cursor.[1] (ESCn deletes previous lines entirely.)
^ DEL LINE	Deletes (or restores) the last character or text string that has just been searched for in a search or restore command. For example, SHIFT SRCH abc SRCH will execute a search for the text string "abc" and position the cursor just after the text "abc". If ^ DEL LINE is then executed, the text string "abc" will be deleted. Pressing ^ DEL LINE again will restore the text just deleted.
RESTORE	Inserts text saved by the SAVE key at the cursor's position.
→	Moves the cursor one or more characters to the right.[1]
SHIFT →	Moves the cursor to the end of the line or to the end of the nth line below the current line (with ESCn).[1]
NEXT	Outputs current text down to the cursor, clears the buffer, and inputs the next text buffer load.[1]

GenRad MACRO Commands

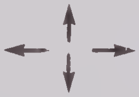
General

Once TECO has been entered, a series of GenRad TECO commands are available for use. The DEC TECO commands described in para 11.7 can also be used. All commands are entered via the keyboard and echoed on the display terminal. An additional series of keys on the right-hand keypad and on the upper right section of the keyboard allow many text editing jobs to be handled with a minimum of effort. The following paragraphs define a useful subset of the various GenRad editing commands available and also list some of the more useful DEC TECO commands that can be used.

GenRad TECO commands can be invoked either by pressing the ^ and another character key (such as ^A to return to the start of a line) or by pressing specific keypad keys (such as 3 to return to the start of a line). DEC TECO commands can be issued whenever DEC TECO itself is entered, as evidenced by an asterisk (*) display, or from the GenRad Macro by pressing the PF2 key.

Keyboard Arrow Commands

A series of single-key commands controlling cursor movement can be entered on the arrow keys (top right) of the main keyboard.



These keys move the cursor in the directions indicated. They can be preceded by ESCn to perform operation n times. ESC ← and ESC → start the cursor moving continuously to the left or right. ESC ↑ and ESC ↓ move it up or down 24 lines.

11.6 VT10X FUNCTION KEYPAD COMMANDS

The keypads shown in Figure 11-5 and 11-6 illustrate the single-key commands that can simplify any editing task for the VT10X TECO and ED2 editors, respectively. The function of each key is given in Table 11-7.

Table 11-7
VT10X FUNCTION KEY COMMANDS FOR VT10X TECO AND ED2

Function Key	Function
0	Advances the cursor one line. The cursor is positioned at the beginning of the line. The 0 key has the same function as N.[1]
ESC 0	In TECO, forward scroll (continuous advance) until halted when any non-special keypad key is pressed.
PF1 0	In ED2, move cursor down 22 lines.
1	Moves the cursor to beginning of the text buffer (same as ^ T).

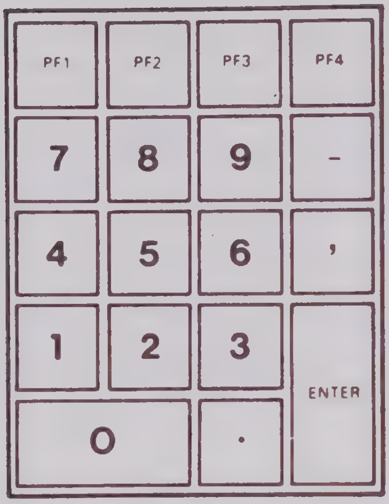


Figure 11-5. VT10X Function Keypad for Teco.

Table 11-7 (Continued)
VT10X FUNCTION KEY COMMANDS FOR VT10X TECO AND ED2

Function Key	Function
ESCn 1	In TECO, moves the cursor to the nth character from the top of the buffer.
PF1 1	In ED2, change the direction of movement for commands that involve cursor movement, etc. Will cause some commands that move forward (down) to back up.
2	Moves the cursor to the end of the current text buffer (same as ^Z).
PF1 2	In ED2, change the direction of movement for commands that involve cursor movement, etc. Will cause some commands that move backwards (up) to move forward (down).
3	Moves the cursor to the start of the current line (same as ^ A).
PF1 3	In ED2, move cursor to the end of the line.
4	Backs up the cursor one line. The cursor is positioned at the beginning of the line (same as ^ H).[1]
PF1 4	In ED2, moves cursor up 22 lines.
5	Deletes the character at the cursor (same as ^D).[1]

---

[1] Can be preceded by ESCn in TECO, or PF1n (gold key) in ED2 to perform operation n times.

```
 _____________________________________________
¦  'PF1'   ¦  'PF2'    ¦  'PF3'    ¦  'PF4'   ¦
¦          ¦  Select   ¦  *Cut     ¦  Help    ¦
¦  <GOLD>  ¦<Command>  ¦ <Paste>   ¦  <Find>  ¦
¦__________¦___________¦___________¦__________¦
¦   '7'    ¦   '8'     ¦   '9'     ¦   '-'    ¦
¦Open line ¦   Page    ¦  Quote    ¦ Adv Word ¦
¦<Append>+ ¦  <Fill>   ¦*<Append>  ¦<Back Wd> ¦
¦__________¦___________¦___________¦__________¦
¦   '4'    ¦   '5'     ¦   '6'     ¦   ','    ¦
¦ Up line  ¦ Del Ch->  ¦Delete Wd  ¦Srch/Repl ¦
¦<Up 22 L> ¦<Rest Ch>  ¦<Rest Wd>  ¦<Rpt S/R> ¦
¦__________¦___________¦___________¦__________¦
¦   '1'    ¦   '2'     ¦   '3'     ¦          ¦
¦Top of Pg ¦  Bottom   ¦StartOf L  ¦ 'ENTER'  ¦
¦<Backup>  ¦<Advance>  ¦<EndOf L>  ¦          ¦
¦__________¦___________¦___________¦          ¦
¦         '0'          ¦   '.'     ¦Enter Arg ¦
¦      Down line       ¦Rpt ^P/^V  ¦ Search   ¦
¦  <Down 22 lines>     ¦<Reset G>  ¦<Ch Case> ¦
¦______________________¦___________¦__________¦
```

Figure 11-6. VT10X Function Keypad for ED2.

Table 11-7 (Continued)

Function Key	Function
PF1 5	In ED2, restore the character just deleted.
6	In TECO, if the previous edit command was a deletion command, restore the deleted text and return the cursor to its former position. Otherwise, delete the spanned text block. The spanned text is the most recently located or inserted text in the buffer referenced by a cut, paste, search, get, or exchange command, or by a restoring 6-key command. In ED2, delete a word to right of cursor.
ESC 6	In TECO, moves the cursor to the location where it was prior to the last operation. For example, if text has been saved using the PF1 key, this command would position the cursor at the location just before the saved text.
PF1 6	In ED2, restore the word just deleted.
7	Opens one line by inserting a carriage return and linefeed sequence below the cursor. This moves the remaining text down one line.[1]

---

[1] Can be preceded by ESCn in TECO, or PF1n (gold key) in ED2 to perform operation n times.

Table 11-7 (Continued)

Function Key	Function
PF1 7	In ED2, appends 1 line of text to whatever text is currently in the PF3 buffer.
8	In TECO, writes out the entire text buffer to an open output file, clears the buffer, and then loads the next buffer from the input file for edit. The 1 and 2 keys can be used to locate the limits (start and end) of each buffer load. When the buffer is empty, an EOB character (diamond shaped) is displayed. It is &not possible to return to the first buffer load again without first closing and then re-opening the output and input files. This can be done by pressing ESC <TAB>. In ED2, advance the cursor to the next form-feed.
PF1 8	In ED2, fill in the selected range of text. This command places a maximum amount of text on each line without breaking words.
9	Inserts the next character typed. Used to insert non-printing characters in the text, such as control characters, that would otherwise be interpreted as editing commands.
ESC 9	In TECO, marks the current cursor position for use with other commands to establish and operate on a text block. To move the mark, press the ESC 9 keys at the new position. When the ESC 9 keys are pressed and the cursor is at the marked position, the mark is removed and the bell sounds. The TECO macro always removes the mark after it has been used.
PF1 9	In ED2, appends a marked range of text to whatever text is currently in the PF3 buffer.
PF1 and PF3	In TECO, saves the line or lines down from the current cursor as text in a register and inserts them anywhere in the file upon command. The PF1 key allows text to be saved in a register. The PF3 key inserts the text into the file at any cursor position. The cursor is positioned at the first line or lines to be saved, and then the PF1 key is pressed. The cursor will be re-positioned to the line after the last line to be saved as the command is executed. The text saved is not deleted from the text buffer. Next, the cursor can be repositioned to the area that is to receive the saved text. When the PF3 key is pressed, the text will be entered just above the cursor position. For example, ESC 4 PF1 saves the 4 lines from the cursor position down. The cursor can then be re-positioned to that point in the file that is to receive the saved data. When the PF3 key is pressed, the lines are inserted directly above the new cursor position.[1]

---

[1] Can be preceded by ESCn in TECO, or PF1n (gold key) in ED2 to perform operation n times.

Table 11-7 (Continued)

Function Key	Function

	This command is ideal for saving and restoring small segments of text (usually 10 lines or less at one time). It should be noted, however, that text entered into the text buffer and not deleted can fill the buffer to a point where a
	`?MEM MEMORY OVERFLOW`
	`*`
	message will be generated. If this should occur, the TECO editor itself will be entered as evidenced by an asterisk (*) display. To return to the GenRad macro and continue the editing session, the user enters
	`*MI$$` (where $$ indicates press the ESC key twice.)
	Next, the cursor should be positioned just below the text to be output, and a Y command issued to output text from the buffer. If the message continues to occur, or if the GenRad macro cannot be entered through use of the MI command, a user should exit TECO (EX ESC ESC) and then re-enter the program again.

PF2	In TECO only, this key allows DEC TECO commands to be entered directly from the GenRad macro. For example, pressing the PF2 key results in the display of a `Command:` message at the top of the display. A user then enters the DEC TECO command and executes it by pressing the ESC key twice. ESC is echoed as a dollar sign ($) each time that it is pressed. Some useful TECO commands that can be entered are described in para 11.7 In ED2 only, places a mark that defines one end of the 'select' range used in moving blocks of text. The other end of the range is defined by the current cursor postion.
PF3	In ED2, saves the currently marked text for insertion at a later time with the PF1 PF3 keys. The text is deleted from the buffer.
PF4	In ED2, displays a menu type HELP page that lists the major functions of the keypad keys and of characters.
PF1 PF2	Allow ED2 line mode commands to be executed from the GenRad character mode. For example, specifying `Command: SHOW BUFFER` will display a listing of all user-created buffers.

Table 11-7 (Continued)

Function Key	Function
	Execute search, search/replace, and find commands entered from keypad. Repeat search and search/replace. Enter line mode commands submitted from screen editing mode.
PF1 PF3	In ED2, "paste" the "cut" text from the paste buffer into the file just above the cursor position. Next, move the cursor to the location that is to receive the "cut" text. Text will be inserted just above the cursor position.
PF1 PF4	Move to the ED2 buffer specified. Text can be stored in several buffers (refer to para 11.8.3). For example, entering the name ABC would display the contents of the ABC buffer if one existed, and create an empty ABC if it did not previously exist.
PF1 -	In ED2 moves cursor 1 word to the left. (Same as ^J.)
PF1)	In ED2 repeats the most currently executed search and replace command.
PF1 ENTER	In ED2, change case.
PF1 .	In ED2, resets marked cursor position and cancels the selected range. This function is provided to allow the user to cancel the select range without executed a command. Normally, the range selection is automatically cancelled after execution of a key that operates on the range. For example, the range is cancelled automatically after execution of a cut operation.
-	In ED2 only, move cursor 1 word to the right.
,	Search and replace as described for ^ V command (Table 11-2).
ENTER and . (DOT)	When the ENTER key is pressed, a SEARCH prompt occurs. Enter the search text string followed by ESC ESC or the . (DOT) key. The search will then be initiated from the current cursor position forward until the first occurrence of the text is found. The cursor is positioned just after the text string, if located; otherwise, a CAN'T FIND "text" message is generated and the cursor is returned to its original position prior to the search. To continue the search for the next occurrence of the text string, press the . (DOT) key.
PF1 & D	In ED2 only, redefine command key. This function is described in detail in para 11.8.5.
PF1 & I	In ED2 only, include file. This performs the same function as the ED2 line editor. INCLUDE command described in Table 11-9.

Table 11-7 (Continued)

Function Key	Function
PF1 & L	In ED2 only, the cursor is located where it was last positioned in the buffer most-recently accessed prior to the current one. That prior buffer is then displayed.
PF1 & M	In ED2 only, the cursor is located where it was last positioned in the MAIN buffer and MAIN is displayed.
PF1 & P	In ED2 only, the current buffer is written to the output file in "print" format. In print format characters such as ESC are shown in the file as <ESC>.
PF1 & W	In ED2 only, the current buffer is written to the output file.
PF1 & X	Writes out the rest of the file being edited and returns to the EDIT mode options page or to an Edit-On-Errors ATG run.
ESCn ENTER and . (DOT)	When the ENTER key is pressed, a SEARCH prompt occurs. Enter the search text string followed by ESC ESC or the . (DOT) key. The search will then be initiated from the current cursor position forward until the nth occurrence of the text is found. The cursor is positioned just after the text string, if located; otherwise, a CAN'T FIND "text" message is generated and the cursor is returned to its original position prior to the search. To continue the search for the next occurrence of the text string, press the . (DOT) key.
DELETE	Delete one character to the left of the cursor.[1]
PF1 ^E	Quit ED2 and escape from ATG edit-on-erros loop without creating a new file.

---

[1] Can be preceded by ESCn in TECO, or PF1n (gold key) in ED2 to perform operation n times.

11.7 DEC TECO COMMANDS

TECO Commands may be issued from the GenRad macro for the console keyboard by pressing the

SHIFT HELP

keys, entering the TECO command, and then pressing the ESC key twice. All TECO commands are executed by pressing the ESC key twice.

TECO commands may also be issued by entering TECO directly through activation of the C or G keys. When TECO is entered, an asterisk (*) is displayed. At this point, the TECO command may be issued and executed. Type MI ESC ESC to return to the GenRad macro from TECO. Some of the TECO commands listed in the following paragraphs may prove useful for text editing. A complete set of TECO commands is included in the DEC TECO-11 User's Manual.

Inserting a New File

The following is a typical example of how TECO (* prompt) can be used to insert a new file (e.g., FILENEW.TPG) into an existing file (e.g., FILEOLD.TPG). It is meant only for illustration, since the same function is already provided by the command ESC i.

TECO Command	Description
*ER FILEOLD.TPG$	Read the existing file.
*N insert-position$	Search the existing file for the position at which to insert the new file. An N-search will search the entire file for the character string indicated, and place the pointer at that position.
*0,.P$	Write out the file, from the beginning to the cursor position.
*HK$	Kill (delete) the buffer.
*ER FILENEW.TPG$	Open the new file to be inserted.
*N^N^X$	Move the pointer to the end of the new file. Note: the caret (^) indicates the key.
*ER FILEOLD.TPGn$;	Re-read the file to which the insert is being made. Include the version number if FILEOLD and FILENEW have the same name.
*-1ED$	Enable the following search.
*_insert-position$	Search for the position at which to insert the new file and place pointer at that position.
*0,.K$	Kill (delete) the buffer from the top (beginning) to the pointer positon,
*EX$	Close input and output files and exit TECO.

Exiting the Macro

After the editing session has been completed, press the ESC and X keys to return to the GenRad Edit Mode Options Page. This command automatically outputs any text in the text buffer, closes the output file, and exits back to the GenRad Edit Mode Options Page.

It may be desirable for a background user to use TECO while the foreground user is testing boards. In this case it is advisable to remain in the DEC TECO rather than the GenRad TECO Macro. Control is passed from the Macro to DEC TECO by pressing G keys. When DEC TECO is active and ready to receive commands, the TECO prompt (*) will appear on the display. All DEC TECO commands are executed by pressing the ESC key twice. Type MI ESC ESC to return to the GenRad Macro from TECO. DEC TECO commands can also be issued from the GenRad Macro by pressing the PF2 key, entering the TECO command, and then pressing the ESC key twice to execute the command. Some of the TECO commands listed in Table 11-8 may prove useful for text editing. A complete set of TECO commands is included in the DEC TECO-11 User's Manual.

NOTE

Almost all keyboard key combinations execute some TECO command. Always check the command string echoed on the Display Terminal before executing the command. Incorrect commands can be deleted by pressing the DELETE key or U keys.

Table 11-8
SOME USEFUL DEC TECO COMMANDS

SEARCH COMMANDS Searches start from the current cursor position. Two types of searches are available. The S-type searches only in the current text buffer, while the N-type searches in the entire file. Type MI to return to the GenRad macro from DEC TECO when a search is unsuccessful. Successful searches position the cursor directly after the text being sought.

ALL COMMANDS The $ symbol shown in the command lines that follow must be generated by the ESC key, not the dollar ($) key. A number, denoted by n in the command string, can be used in many commands to indicate a quantity other than the default value of one.

Command	Purpose
nStexT$	Search for the nth occurrence of the text listed, but do not search beyond the end of the current text buffer.
-nStext$	Same as above, but search backwards in current text buffer.
nNtext$	Same as above, but search the entire file, inputting and outputting text until the search string is located.
nFStext1$text2$	Search for nth occurrence of text1 and replace it with text2.
nFNtext1$text2$	Same as above, but search may continue beyond the current buffer. Each time the end of a buffer is reached, it is copied to the output file and a new page is read in from the input file.
<FStext1$text2$>	Repeat the search and replace operation until all occurrences in the current text buffer have been replaced.
<FNtext1$text2$>	Same as above, except that all occurrences in the entire file are replaced.
MI	Re-enter the GenRad macro from DEC TECO.
J	Moves cursor to top of text buffer.
ZJ	Moves cursor to bottom of text buffer.
nC	Advances cursor n positions to the right.
-nC	Advances cursor n positions to the left.
nL	Advances cursor n lines down

Table 11-8 (Continued)
SOME USEFUL DEC TECO COMMANDS

Command	Purpose
-nL	Advances cursor n lines up.
Itext$	Insert the text just prior to the cursor position.
nK	Deletes n lines of text.
nD	Deletes n characters at and to the right of the cursor.
-nD	Deletes n characters to the left of the cursor.
P	Outputs the text buffer, clears the buffer, and loads the buffer with the next segment from the input file.
EC	Outputs any remaining text and closes the output file. Does not exit TECO.
EK	Deletes the entire output file.
EF	Closes the output file. Does not output any remaining text.
ER Dev:Name$	Opens the file specified for input.
EB Dev:Name$	Opens the file specified for input and output.
EW Dev:Name$	Opens the file specified for output.
EX	Outputs any remaining text, closes the output file, and exits back to MCR (> prompt) or to the monitor.
A	Appends the next page of text from the input file to the text that is currently in the text buffer. The formfeed at the end of the existing page is deleted so that the buffer becomes a single page for output.
PW	Write the current contents of the text buffer to the output file and insert a formfeed at the end of it. Nothing is deleted from the buffer and the cursor is not moved.

Searches do not differentiate between upper and lower case characters when TECO is initially entered. To change this feature, press the G keys and enter one of the following commands in response to the asterisk prompt (the ^X is entered by pressing the and X keys simultaneously; the $ symbol means the ESC key is pressed):

*-1^X$$	Search for an exact match of upper and/or lower case characters.
*0^X$$	Do not differentiate between upper and lower case characters during a search.

and then enter the command

*MI$$ Return to GenRad Macro.

All DEC TECO commands are executed by pressing ESC twice. Once invoked, these commands remain in effect for all search commands (GenRad and DEC) until changed or until TECO is re-entered.

DEC TECO Search and Replace Commands

Commands of the form <FSText1$Text2$;> or <FNText1$Text2$;> are extremely useful for replacing repetitive text in a file. The <FN> command replaces all occurrences of text string 1 with text string 2 in the file. For example, if the word L is to be changed to F throughout the file, then the <FN> search and replace command can be issued once (with the cursor positioned at the top of the file) to change the entire file. Upon completion of the search operation, the command ESC F can be issued to return to the start of the file and continue the editing session.

The <FS> command replaces all occurrences of text string 1 with text string 2 in the current text buffer only (not the entire file).

The N search command is another handy tool. Followed by other TECO commands, such as C, D, N, and K, it can be used to carry out search and replace operations that cannot be performed with <FN>. For example,

```
<N%value$LKI/*- - - - */$>
```

searches the file for all occurrences of "%value," deletes the next line of text and inserts "/* - - - - */."

DEC TECO Q-Register Commands

TECO Q-Register commands can be issued to save text in registers. The text can then be output at some other cursor location in the file. The DEC TECO commands that save and restore text are:

nXq	(saves text)
Gq	(restores text)

where: n Represents the number of lines of text to be saved from the current cursor position down, with the cursor positioned at the start of a line).

Xq Is the command to save text in the Q-Register named q. Q-registers with single letter names A through Z, except J, are available. Single number 0 through 9 registers are also available; however, the GenRad macro resides in most of them (registers 4 through 9), and therefore these registers cannot be used.

Gq Is the command to restore text in Q-Register q just above the current cursor location (with the cursor positioned at the start of a line).

The following procedure illustrates how 24 lines of text can be saved in a register and restored at some other location in the file while operating in the GenRad macro.

a. Position the cursor at the start of the line or lines to be saved.

b. Press the ^C keys. When the asterisk prompt occurs, issue the following command

```
*24XA$MI$$
```

to save 24 lines down from the current cursor position in Q-Register A.

NOTE

To avoid a MEMORY OVERFLOW condition caused by a full text buffer, it is advisable to position the cursor at some point above the area that is to receive the saved text and execute a Y command to clear the buffer of some text _before_ adding the saved text.

c. Re-position the cursor to the new location in the file that is to receive the saved text string. The text will be entered just above the cursor when positioned at the start of a line.

d. Press the ^C keys, and issue the following command in response to the asterisk prompt to enter the saved text:

```
*GA$MI$$
```

TECO Error Messages

TECO error messages take the form

```
? (Code)  (Message Description)
```

For example, a memory overflow condition would result in the generation of a

```
?MEM  MEMORY OVERFLOW
*
```

message. A TECO search command that did not locate the text string being searched for would generate a

```
?SRH SEARCH FAILURE "String"
```

message. In most cases, DEC TECO will be entered when the error message is generated, as evidenced by the asterisk (*) display.

To re-enter the GenRad macro, issue the command

```
*MI$$
```

only if a MEMORY OVERFLOW error message has not been generated. If a MEMORY OVERFLOW error has occurred, issue the following set of commands

```
*04,0X7$$
*0,0X3$$
*MI$$
```

to clear the TECO 3 and 7 Q-Registers before re-entering the GenRad macro. Next, the cursor should be positioned just below the text to be output, and a ESC W command issued to output text from the buffer. If the error condition persists, or if the GenRad macro cannot be entered through use of the MI command, exit (EX command) and then enter TECO again.

If a

```
?NFO NO FILE FOR OUTPUT
*
```

message is displayed, issue a

```
*EWDEV:FILENAME.EXT$$
```

command to define the missing output file and device location. To exit TECO, issue the command:

```
*EX$$
```

RSX-11M System I/O Handler error codes can also occur. For example, attempting to output a file to a full device will generate a

```
? 024 Device full
```

error message.

Some of the more common error messages are listed in Appendix B. For a complete set of DEC I/O error codes refer to the IAS/RSX-11M I/O Operations Reference Manual available from Digital Equipment Corp.

11.8 ED2 EDITOR

11.8.1 Introduction

ED2 is Version 2.0 of DEC's[1] EDT interactive text editor. The editor operates in both screen editing and line editing modes. The GenRad version automatically enters screen editing mode at startup to provide users with the advantages of keypad editing on a VT10X terminal. The functions of the numeric keypad, on the right side of the VT10X keyboard, and control character commands are redefined in ED2 to be consistent with other GenRad editors. Though the user can perform all editing functions from the screen mode, line mode editing can be used either by submitting line mode commands from the screen mode or by enabling the ED2 line mode.

Features

The ED2 editor imposes considerably less demand on the CPU than does TECO.

ED2 operates in both a screen mode and a line mode for editing. Screen editing is the default.

The text buffer provided by ED2 is the largest of the 3 editors. This is accomplished by saving text in a "virtual" memory buffer, which makes it appear that the entire file is in memory at one time. This is really not so, but it does allow the user to move from the start of the file to the end of the file and back to the start again without having to write out the file, leave the editor, and call the file again in another editing session.

---

[1] Digital Equipment Corp., Maynard, MA 01754.

Multiple text buffers can be opened to hold other files or sections of files. Files other than the main file being edited can be input (INCLUDEd) by the editor into the main text buffer or any auxiliary buffer the user chooses. Any buffer or section of a buffer can be written (WRITE) to any system disk device and saved under any name the user assigns.

Macros can be defined and written, which can then be used to perform a sequence of editing tasks by issuing a single command to call the macro.

Finally, ED2 keeps a journal file (extension .JOU) of all the editing commands which are entered during an editing session. When an editing session is terminated normally, with the EXIT or QUIT command, the journal file is automatically deleted. If, however, the editing session terminates abnormally, due to a system "crash", the journal file is not deleted and can be used in a "recovery" (/RECOVER) operation by ED2 to reconstruct the file to the condition it was in just before the termination.

Requirements

ED2 commands and functions are defined and pre-enabled by the GenRad supplied initialization file E.EDT, which must be on the user's default device. If this file does not exist, the editor will be enabled in the regular DEC ED2 line mode, indicated by an asterisk (*) prompt. To leave the editor from this mode without creating an empty file, enter the following commands to the * prompt.

```
*QUIT
```

The following paragraphs present the ED2 screen editing commands and line mode editing commands used from a VT10X terminal. Enough information is provided to prepare the user to perform most standard editing sessions rapidly and efficiently. Advanced users of the editor should refer to the DEC EDT Editor Manual (AA-J726A-TC) for a complete description of all available ED2 commands.

11.8.2 Accessing ED2

ED2 is normally accessed through the EDIT page of the monitor. Entry command lines to access ED2 from the MCR prompt (>) differ, depending on the type of editing job to be accomplished. (MCR is active when a monitor SHUt command is executed.) The following command lines can be used to enter ED2 and accomplish the indicated tasks.

Task A. Create a new file

```
>ED2 SC1:IOBOARD.CKT
```

When the RETURN key is pressed, ED2 is entered and the message

```
Input file does not exist
```

is displayed momentarily, and an End Of Buffer symbol ([EOB]) appears at the top of the screen. The cursor is initially placed on the same line as the EOB symbol, but when text is entered the EOB symbol is automatically moved down a line. Once the editing session has been completed, a new output file (SC1:IOBOARD.CKT) with the version number 1 will be written.

Task B. Edit an existing file without changing the file name or device location.

```
>ED2 SY:OVPTEST.TPG
```

When the RETURN key is pressed, ED2 is entered, and the first screen-sized page (22 lines) of the file is displayed. The cursor is positioned at the top (first line) of the file. Text can now be entered, or editing started, using the procedures described in para 11.8.4. Once the editing session has been completed, a new output file will be created on the same device, with the same name and extension, but with a version number incremented by 1.

Task C. Edit an existing file but change the output file name and/or device location.

```
>ED2 AL:FIXTEST.TPG=SY:OVPTEST.TPG
```

The operation of ED2 is the same as described for Task B above but, when the editing session has been completed, an output file will be created with the new file specification (AL:FIXTEST.TPG).

Task D. Recover a file after abnormal termination of an editing session (e.g., after a system "crash").

NOTE

The following procedures should be used only to recover from a system "crash" that occurred during the last previous editing session. The journal file kept by ED2 applies to a specific version of the text file. If another version of the text file is created before RECOVER is run, ED2 will apply all editing commands from the journal file to the later version, often with unpredictable (though sometimes humorous) results.

Specify the appropriate command line described under tasks A through C, adding the /RECOVER specification to the command line. For example,

```
>ED2 SC:BIGBOARD.CKT/RECOVER
```

starts a recovery operation when the RETURN key is pressed. Recovery operations allow users to recover most (up to the last few key strokes) of the text that would otherwise be lost if a sudden power failure or system "crash" occurs while the file is open in the editor. Details for running recovery are in para 11.8.7.

11.8.3 GenRad Keypad Editing (Screen Mode)

The VT10X Terminal Keypad editing functions described in the DEC EDT Editor Manual (AA-J726A-TC) have been re-defined by GenRad to provide compatibility with other GenRad editors when operating in the screen editing mode. The GenRad VT10X Terminal layout is shown in Figure 11-6 and described in para 11.6.

Characters shown in single quotes in the Figure 11-6 (e.g., 'PF1') are the actual VT10X keypad markings. Two different levels of commands can be issued from the same keypad either by pressing a single key to execute one command or by pressing the same key after pressing the PF1 key. Commands shown at the top of each key (for example, DOWN LINE on the 0 key) are executed when that key is pressed by itself. Commands shown on the bottom of each key, such as <DOWN 22 LINES>, are only executed when the PF1 <GOLD> key and then the other key are pressed. For example, pressing the large 0 key at the bottom of the keypad moves the cursor down 1 line. Pressing first the PF1 key and then the 0 key will move the cursor down 22 lines.

Press the PF4 key to display the basic commands, and the ENTER key to terminate the display.

In addition to the keypad, a series of commands can be issued from the terminal keyboard. Many of these commands provide similar functions to the keypad and are compatible with other GenRad editors. These commands are summarized in paragraphs 11.3 and 11.4.

11.8.4 Basic ED2 Editing

Once ED2 is entered using the procedures described in para 11.8.2, the editing session can begin. The GenRad screen editing mode is automatically entered initially and the first page of the file to be edited (or the EOB symbol if a new file) is displayed.

Selecting Editor Modes

Normally, the ED2 screen editing mode is entered automatically when the GenRad supplied initialization file is present on the system default device. If the ED2 line mode is ever entered, the asterisk prompt

```
*
```

is displayed. The ^G or ^C commands (hold down the key while pressing the G or C key) can be used to enter the ED2 line mode at any time. To re-enter the screen editing mode, enter the command

```
*C          (the C stands for CHANGE)
```

and press the RETURN key. The RETURN key is used to enter line mode commands when the editor is being used in line editing mode.

When line mode commands are submitted from the screen editing mode Command: prompt (press PF1 and PF2), the keypad ENTER key must be used instead of RETURN. If RETURN is pressed, the command is not entered and the characters ^M appear at the end of the command. Use the DELETE key to remove the ^M and then press ENTER.

Cursor Movement

Effective screen editing depends on the ease with which the cursor can be repositioned on the screen. Commands which move the cursor by a number of characters or lines (the default is 1 character or line) can be repeated by first pressing the GOLD (PF1) key and entering the number of times to repeat the movement. The number is displayed in the lower left of the screen when it is entered.

Search For Text

The search feature is a handy tool for locating text to be changed. Searches are conducted from the cursor position to the end of the file (or the beginning of the file if the direction is set to "Backup"), so the cursor must be positioned before the search target (or after, if direction is "Backup"). The following commands can be used to locate text.

Text Search

ENTER key (on keypad). Displays

```
Search:
```

Enter the text string to be search for and press the keypad "." key to execute the search. The cursor is positioned just after the text string searched for.

NOTE

A string containing a single quote (tick) or apostrophe (') cannot be found using the screen editing mode keypad search command. To search for a string containing this character, use the line mode FIND command described in para 11.8.6.

Search and Replace Text

, key (on keypad). Display

```
Replace:
```

Enter the text string to be replaced ("string one"), and press the "," key again. The message

```
Replace: string one  With:
```

appears. Enter the new text ("string two"). The message should now read

```
Replace: string one  With: string two
```

Press the "." key to replace the text.

Moving Lines of Text

Lines of text may be moved from one location to another within a file. The following procedure can be used for this purpose:

a. First, position the cursor at the start of the line to be moved.

b. Next, press the "Select" key (PF2) to mark the start of the range of text. Now use the arrow keys or control commands to advance the number of lines to be "cut." The lines to be cut will appear in reverse video.

c. Press the "Cut" key (PF3) to remove the lines in the selected range from the screen and save them in the paste buffer.

d. Next, re-position the cursor to the place in the file where the "cut" text is to be re-inserted.

e. Press the "Paste" key (second function of the PF3 key, entered by first pressing the PF1 <GOLD> key and then the PF3 key) to "paste" the lines into the file. The cut text will be entered on the line directly above the new cursor position. Other methods of moving text are described in para 11.8.6.

Text Entry and Deletion

Text is normally entered by positioning the cursor at the desired location in the file and then typing the text directly on the terminal keyboard. New text is immediately visible on the terminal display as it is entered and the cursor is advanced. If an incorrect character is inserted, the keyboard DELETE key can be pressed to remove it. A new line for text can be opened by pressing the keypad 7 key. If a mistake is made, the ^K keys can be pressed to delete the entire line. If the wrong line has just been deleted, the ^X keys can be pressed to restore the line.

Text may be deleted using any of the following methods, as appropriate:

1. Press the DELETE key to delete a character to the left of the cursor.

2. Press the ^D keys to delete a character at the cursor.

3. With the cursor positioned at the start of a line, press the ^K keys to delete that line. When the cursor is not at the start of a line, ^K deletes text from the cursor position through the end of the line (including the CR-LF).

Examples of Text Entry and Deletion

1. Entering and Editing a New File. When ED2 is entered, a new file can be input directly from the terminal keyboard. As text is entered, it can be viewed directly on the terminal display. If errors are noted, the cursor can be repositioned where necessary, and the DELETE, ^D, ^K and ^U keys can be used to delete text. For example, assume that the following text has been entered:

   "The keyboard has siveen light emitting diodes (LEDs) that are committed to the ON LINE/LOCAL function."

   After completing the second line, it is noted that seven (underlined) has been misspelled. The following actions can be taken to correct the text:

   1. Press the ^H keys to move the cursor up 1 line and to the start of that line.

   2. Press the "," keypad key. A

      Replace:

      message should occur. Enter the text string to be replaced.

      Replace: siveen

and again press the "," keypad key. The message

```
Replace: siveen With:
```

will occur. Enter the new text string

```
Replace: siveen With: seven
```

and press the "." keypad key to execute the search and replace command and correct the misspelled word.

2. Editing Existing Text. This paragraph lists some of the basic commands that can be used to edit text. To edit text efficiently, a user should also be familiar with the complete set of GenRad keypad editing commands described in para 11.6, the ED2 line editing commands described in para 11.8.6, and the DEC EDT Editor manual (AA-J726A-TC).

   When the GenRad screen editing mode is first entered, the first page of the file is automatically displayed, and the cursor is positioned at the top of the file. The easiest way to locate text that requires change is to search, with the ENTER key, for the text string needing modification. For example, if you press the ENTER key, the prompt

   ```
   Search:
   ```

   will appear. Enter the text string to search for and press the . (dot) key to execute the search command. If the text string is located, the cursor will be positioned just after the string. At this point, editing can begin.

3. Exiting ED2. The command to exit from ED2 is a line mode command, so it must be entered from the asterisk prompt or by using the <Command> keypad function key. To exit from the editor, press the PF1 key and then the PF2 key. The prompt

   ```
   Command:
   ```

   appears at the bottom of the screen. Enter one of the following commands to terminate the session and re-enter the monitor level.

   ```
   Command: EX
   Command: QUIT
   ```

   EXIT writes the contents of the MAIN buffer to the output file and leaves the editor. QUIT leaves the editor without writing an output file.

11.8.5 Defining Key Functions

When a series of keypad commands is used repetitively, it is often convenient to define a key on the keyboard to perform the commands. ED2 provides a means to allow a series of commands to be learned and assigned to a /key or <GOLD>/key pair. The user does not have to know the mnemonic screen-editing commands of ED2 to use this feature, though a familiarity with the keypad functions is necessary.

To use the Define Key feature, press the keypad <GOLD> key (PF1) followed by the letter D key. ED2 will then prompt for the key to define and the definition.

In the following example, assume we have a file with an 8-digit line number and 8 spaces at the beginning of each line. To remove the 16 leading characters from a line, place the cursor at the start of the line and press the Delete Char key (keypad 5) 16 times. This method, though, requires 17 key-strokes for each line. If the file in question is 1000 lines long, this method would be time-consuming and tedious.

To perform the same function more efficiently, the 17 key-strokes can be learned using the Define Key feature. The following example defines the <GOLD> U key to delete the leading 16 characters of a line.

```
User enters:        <GOLD> D
ED2 responds:       Press the key you wish to define
User enters:        <GOLD> U
ED2 responds:       Now enter the definition, terminated by ENTER
```

The definition requested by ED2 can be typed directly from the keyboard by using the keypad mnemonic commands or by pressing the keypad key itself. The latter method is generally easier to use. The syntax rules for definitions require certain commands to be grouped into units with parentheses (the repeat Delete Char in the description which follows is an example of this). Command definitions containing a period (.) as the last character will execute immediately when the defined key is pressed. If no period is present, the keypad ENTER key must be pressed after the defined key to execute the command.

To enter the definition, first type a left parenthesis which will enclose the entire command string. Then, since we wish to delete 16 characters and the Delete Char key accepts repeat numbers, the command will include the repeat count. Press the <GOLD> key and type 16 (using the keyboard keys). Next, since the repeat is to apply to the Delete Char command, the command must be set off as a group. Type a left parenthesis, press the Delete Char (5) key, and type a right parenthesis. After deleting the characters from the current line, the cursor should be moved to the next line. Enter the keypad 0 key (Down line) to do this, which also insures that the cursor will be placed at the start of the line. Close the command string with a right parenthesis and a period. The last character must be a keypad period as shown. The command string should look like this:

```
(16(+DC)+L).
```

Save the definition by pressing ENTER. The definition is assigned to the key, but the command is not executed. To use the new function key, move the cursor to the start of the current line. Then enter <GOLD> U. The leading 16 characters of the current line are deleted and the cursor is placed at the start of the next line. Pressing <GOLD> U again will delete the characters from this line. The number of key-strokes required for each line has been reduced from 17 to 2, but this is still many key-strokes to remove the characters from the rest of the 1000 lines.

To reduce the number even more, the repeat count form we used to perform the delete command can be used also with the newly defined command. To do this, press the <GOLD> key, type the number 1000, and then enter the <GOLD> U command. The leading characters from all the remaining lines in the file will be deleted, and when the end of the file is reached the message "Advance past bottom of buffer" appears. Attempting to repeat a movement or search command beyond the end of the buffer will result in this message, and the movement or search is stopped. By defining a new key function and using repeat counts, the entire 1000 line file can be edited to remove the leading characters with just 20 key-strokes, instead of the 17,000 originally indicated.

NOTE

> When new key functions are being defined, to avoid confusion, use keys which have not already been defined. Also, some keys (e.g., <GOLD>) cannot be redefined.

11.8.6 ED2 Line Editing

The ED2 line mode may be used to transfer, change, or delete large segments of text. The various commands available in the line mode are described in the DEC EDT Editor Manual (AA-J726A-TL). This paragraph highlights some of the more important features available in this mode; it does not, however, describe all the various command formats and procedures available to a user. It is assumed that users have read the DEC manual before they attempt to use line mode operations.

Text Buffers

Text from the file specified when you first enter ED2 is automatically placed in a buffer called MAIN. The editor also automatically creates other buffers during an editing session. For example, a "paste" buffer stores text from editing operations such as CUT.

A user can create other buffers for text storage in addition to the MAIN buffer. For example, a user may wish to create a separate buffer "ABC" and save a file in it that will later be moved in sections to the file being edited in the MAIN buffer.

Unless otherwise specified, files are always input to the MAIN buffer. Text in the main buffer is changed, as required and, after the editing session has ended, the MAIN buffer contents are output to a file. You can, however, create several text buffers during an editing session and manipulate text between them. Some of the commands available for buffer manipulation are described in Table 11-9.

Table 11-9
ED2 LINE MODE COMMANDS

Command	Function
CHANGE	Re-enter the GenRad keypad character mode.
COPY =ABC 30:150 TO =MAIN 180	Copy lines 30 through 150 from the "ABC" buffer and insert them in the "MAIN" buffer just above line 180.
COPY 15 THRU 60 TO 95	Copy lines 15 through 60 of the current buffer to the positions just before line 95. Does not delete the original lines.
DELETE ABC	Deletes the entire "ABC" buffer.

Table 11-9 (Continued)
ED2 LINE MODE COMMANDS

Command	Function
DELETE 20	Deletes line 20 of the current buffer.
EXIT	Exit ED2 and save the contents of the MAIN buffer in a file.
FIND =ABC	Find the "ABC" buffer; the cursor is placed at the start of the buffer.
FIND =MAIN	Find the "MAIN" buffer; the cursor is placed at the start of the buffer.
FIND =MAIN.	Same as FIND=MAIN, except cursor is placed where it last was in MAIN.
FIND 60	Find line 60 of the current buffer.
FIND "tick mark (')"	Find the line containing the string tick mark ('). This search command is used to locate strings containing tick marks (apostrophes), which cannot be found using the keypad search command.
HELP	Displays information about line mode commands.
INCLUDE SC1:PART1.TPG .	Copy SC1:PART1.TPG into the current buffer; the . (period) following the command stands for the current cursor position and must be separated by at least one space. Text is inserted just before the cursor position.
INCLUDE SC1:PART2.TPG =MAIN	Copy SC1:PART2.TPG into the MAIN buffer, appending the text to the end of the buffer.
INCLUDE DY0:FILENAME1.EXT =ABC	Copy FILENAME1.EXT from DY0: into a text buffer named "ABC".
INSERT 50	Insert new text into the current buffer just before line 50. After text has been enter, use ^Z to re-enter the line mode.
MOVE 15 THRU 60 TO 95	Copy lines 15 through 60 of the current buffer to the position just before line 95, then delete lines 15 through 60.
MOVE 1:122 TO =ABC	Copy lines 1 through 122 to the ABC buffer; then delete the lines from the current buffer.

Table 11-9 (Continued)
ED2 LINE MODE COMMANDS

Command	Function
PRINT ALO:FILNAMB.EXT =ABC	Output the "ABC" buffer into FILNAMB.EXT on ALO. The PRINT command inserts form-feed characters to paginate the output. Use PRINT when the file is to be printed LATER on a line printer; otherwise, use the WRITE command.
QUIT	Exit ED2 without saving the contents of the MAIN buffer in a file.
QUIT/SAVE	Exit ED2 without saving the contents of the MAIN buffer in a however, a journal file of edits, however, is saved (refer to para. 11.8.7).
REPLACE 50	Delete line 50 of the current buffer and insert new text at this position. After text has been enter, use ^Z to re-enter the line mode.
RESEQUENCE	Resequence line numbers in the current buffer.
SHOW BUFFER	List the buffers in use during the current editing session. The current buffer (the one being edited) is indicated by the symbol = in front of the name.
SUBSTITUTE/Text1/Text2/WHOLE	Search the current buffer and replace all occurrences of Text1 with Text2.
SUBSTITUTE/Text1/Text2/WHOLE/QUERY	Search the current buffer for Text1. Query the user every time Text1 is located. A user must specify one of the following commands to proceed with the substitution of Text2: Y (Yes) Make substitution N (No) Do nothing A (All) Make all the remaining substitutions without further queries. Q (Quit) Exit the substitution operation.
SUBSTITUTE/Text1/Text2/20:80	Search the current buffer between lines 20 through 80 and replace all occurrences of Text1 with Text2.
TYPE 1:15	Display lines 1 through 15 of the current buffer, with line numbers.

Table 11-9 (Continued)
ED2 LINE MODE COMMANDS

Command	Function
TYPE 10	Display line 10 of the current buffer, with line numbers.
WRITE SC:TOTAL.CKT =MAIN WHOLE	Writes the contents of the MAIN buffer to the file SC:TOTAL.CKT. The range WHOLE is the default and does not have to be included.
WRITE SC:PART1.CKT =ABC	Writes the contents of the ABC buffer to the file SC:PART1.CKT.
WRITE ALO:FILENAME1.EXT	Transfer the contents of the current buffer into the file FILENAME1.EXT on ALO:.
WRITE ALO:FILENAME1.EXT 10 THRU 90	Transfer the contents of lines 10 through 90 of the current buffer into the file FILENAME1.EXT on ALO:.
WRITE DK3:RPACK.ACS SELECT	Write the "select range" of text (the text marked in reverse video in screen editing mode) to file DK3:RPACK.ACS.

11.8.7 File Recovery

ED2 automatically records the edits you make during an editing session in a journal file. During normal editing sessions, the journal file is deleted when you EXIT ED2. If your editing session is terminated abnormally (perhaps due to a system "crash"), the journal file containing your prior edits can be loaded with the RECOVER command. For example,

```
>ED2 DEV:FILNAME.EXT/RECOVER
```

will start a recovery operation (the file DEV:FILNAME.JOU will be opened at the same time). Recovery operations allow users to recover most of the text that would otherwise have been lost because of the system crash.

Journal files may also be saved at any time by entering the following command from the Command: prompt in screen mode or the * prompt in line mode.

```
Command: QUIT/SAVE   (or  *QUIT/SAVE  from line mode)
```

Note, however, that this is not a normal operating practice. Normally, the EXIT or QUIT commands would be used. This command can be used if, for example, a user has to interrupt an editing session in which text from one or more input files has been split among several buffers for reorganization. In this case, EXIT would save only the MAIN buffer, and QUIT by itself would save nothing. A QUIT/SAVE exit from the editor will save in the journal file all the commands which the user has entered to create all the buffers. Opening this file with the /RECOVER command will perform all edits and create the buffers again.

Also, it is important to know that ED2 saves the journal file on the user's default disk device under the user's own UIC. This means if a SET /UIC command is used to work on files under a different UIC and a journal file is created by an abnormal termination of the editing session or with the QUIT/SAVE command, the journal file is written to the user's UIC on his default device, not to the device and UIC he was working on.

11.8.8 File Control

When a file that is not the latest version is being edited without an explicit version specified for output, the edited file will replace the version specified as input. In otherwords, if there are versions 1, 2, 3, and 4 of a particular file and you issue the command: ED2 name.ext;3, the edited version will be version number 3.

11.8.9 Error Messages

Refer to Appendix A of the DEC EDT Editor Manual for a listing of editor error messages.

11.9 GenRad TEX PROGRAM

11.9.1 Introduction

TEX is a text entry program designed to support bulk text entry, with maximum input speed and minimum program size. TEX has no text editing capability other than that which is inherent under RSX-11M. TEX is used for the initial entry of text. A standard text editor, such as ED2 is then used to make corrections.

TEX uses a few simple commands and issues descriptive messages which are suited for a typist who is not a particularly experienced computer user. TEX also features a file save routine to limit text losses caused by system crashes or other malfunctions.

11.9.2 Using TEX

The monitor should be shut down, using the command ABORT, before TEX is used. The TEX program is entered from the MCR prompt (>) by typing the command:

```
>TEX
```

If TEX has not been installed on the system, it may be accessed by typing

```
>RUN DK1:[2,70]TEX        (DM1: on 2271, 2272; FW1: on 2275)
```

TEX will respond with the prompt

```
TEX>
```

indicating that it is waiting for a command. Only the first character of each command needs to be entered for TEX to interpret it correctly. From the TEX prompt (TEX>) the following commands may be entered:

FILE filespec	Causes the specified file to be opened for output. If an output file is already open, that file will be closed prior to opening the new file. If the file is an existing file, output from TEX

is appended to the end of that file. If the file does not already exist, TEX creates a new file.

INSERT	All text entered from the keyboard after this command becomes part of the created file. Z is entered on a line by itself to stop the inserting of text.
SAVE nnn	Causes the output file to be saved every nnn minutes. If a system crash occurs, only the text that was entered since the last save will be lost. If the SAVE command is entered with no number, an immediate save is done. If nnn is 0, the file is not saved, and automatic saving is disabled. The SAVE command can be changed during text entry session.
DONE	Closes the output file and returns to MCR.
HELP	Displays a summary of TEX commands.

TEX has some limited editing functions that can be used after text entry has begun. These functions are primarily invoked via control characters. A summary of TEX editing functions follows:

C (^C)	Aborts TEX and returns to MCR.
I (^I)	Moves the cursor to the next tabstop position.
L (^L)	Inserts a formfeed into the text. TECO recognizes a L as the end of an editing page.
R (^R)	Causes the current input line to be re-typed.
U (^U)	Causes all the characters in the current line to be deleted. The line does not appear to be erased, but a ^U is inserted, and the cursor is moved to the next line.
Z (^Z)	Indicates the end-of-file. If Z is issued while entering TEX>SAVE 5

d. To begin entering text, the typist enters the command:

```
TEX>INSERT
```

and presses RETURN. TEX will respond with the message:

```
[TEX] -- INSERTING....  ENTER TEXT FOLLOWED BY -Z.
```

Text can now be entered until the typist issues a ^Z command. This will cause the program to return to the TEX prompt. To end the TEX editing session, the typist enters:

```
TEX>DONE
```

TEX will respond with the message:

```
[TEX] -- TERMINATING EXECUTION.
```

If you wish to work on another file, use a TEX FILE command, followed by an INSERT command.

Restart the monitor from MCR by typing @2270.

11.9.3 TEX Messages

TEX issues messages to indicate error conditions or to inform the user of what it is doing. A listing of these messages, and their meanings, follows.

[TEX] -- INVALID COMMAND. An invalid command was entered in response to the TEX prompt (TEX>). Only the first letter of a TEX command is needed, but if any subsequent characters of that command are entered, they must be correct.

[TEX] -- IGNORING GARBAGE AFTER COMMAND. An argument was supplied to a command that does not take an argument, and was ignored.

[TEX] -- FILE NOT FOUND. CREATING NEW FILE. This message indicates that the file specified by the FILE command was not found and that a new file will be created after a subsequent INSERT command.

[TEX] -- FILE FOUND. APPENDING TO EXISTING FILE. This message indicates that the file specified by the FILE command already exists, and that any text that is entered after the INSERT command will be appended to the existing file.

[TEX] -- OPEN FAILURE OR INVALID FILESPEC. An invalid file name, invalid device name, or bad syntax was specified in a FILE command.

[TEX] -- NO OUTPUT FILE OPEN. CAN'T INSERT. The user attempted to insert text without specifying an output file.

[TEX] -- INSERTING... ENTER TEXT FOLLOWED BY -Z. This message is displayed after an INSERT command and indicates that text can be entered into the output file.

[TEX] -- FILE BEING SAVED. PLEASE WAIT...
[TEX] -- PROCEED... The first message indicates that the file-saving procedure has begun. The user should wait until the second message appears before proceeding.

[TEX] -- INVALID OR MISSING ARGUMENT. This message indicates that the file name was omitted from the FILE command or that the argument to the SAVE command was invalid (this argument must be an unsigned integer between 0 and 999).

[TEX] -- OBSCURE SYSTEM ERROR, CODE X. ABORTING TASK. This message is displayed as a result of an RSX-11M hardware or software error. Refer to Appendix C, para C.5, for a list of errors.

[TEX] -- TERMINATING EXECUTION. This message indicates a return to MCR as a result of a DONE command or Z being issued in response to the TEX prompt.

Digital Library Devices – Appendix A

Table A-1

List of Digital Devices in System Digital Library Sorted by Type.
(*Are prime devices)
(»Has SpeedPlus capability)

*100160
*10415
*145101
*1488
*1489
*1702
*2101A
*2102A
*2104A
*2111A
*2112A
*2114
*2115A
*2117
2118
*2125A
2141
*2142
*2147
2148
2149
*2167
*2168
*2308
*2316E
*2332
*2364
*25LS07
*25LS138
*25LS139
*25LS151
*25LS153
*25LS157
*25LS158
*25LS160A
*25LS161A
*25LS162A
*25LS163A
*25LS164
*25LS168A
*25LS169A
*25LS174
*25LS175
*25LS181
*25LS190
*25LS192
*25LS193
*25LS194A
*25LS195A
*25LS240
*25LS241
*25LS242
*25LS243
*25LS244
*25LS251
*25LS253
*25LS257
*25LS258
*25LS273
*25LS299
*25LS373
*25LS374
*25LS377
*2608
2616
*2708
*2716
*2732
2758
*2764
*27128
*29600
*29601
*29610
*29611
29612
29613
*29620
*29621
*29622
*29623
*29624
*29625
29626
29627
*29660
*29661
29662
29663
*3101
3602
3604
*3605
*3608
3622
3624
*3625
*3628
*4062A
4116
*4204
*4210
4211
4214
*4246
*4700
5101
5204
*555
6200
6201
6205
6206
*6230
*6231
*6235
*6236
6240
6241
*6250
*6251
*6252
*6253
*6275
6276
6280
6281
6282
6283
*6284
6285
*6286
*6287
*6289
6300
6301
6305
6306
6308
6309
6330
6331
6335
6336
6340
6341
6348
6349
6350
6351
6352
6353
6380
6381
6382
6383
6384
6385
6386
6387
6389
»*68000
*70C95
*70C97
*7400
*7401
*7402
*7403
*7404
*7405
7406
*7407
*7408
*7409
*7410
*74100
*74107
*74109
*74110
*74111
*74114
*74116
*7412
*74120
*74121
*74122
*74123
*74125
*74126
74128
*7413
74132
*74136
7414
*74145
*74147
*74148
*74150
74151A
*74152
74153
*74154
*74155
*74156
74157
*74159
7416
74160
74161
74162
74163
74164
*74165
*74166
7417
*74170
*74172
*74173
74174
74175
*74176
*74177
*74180
74181
*74182
*74184
*74185
74187
*74189
74190
*74191
74192
74193
74194
74195
74196
74197
*74198
*74199
7420
*74200
*7422
*74221
*7423
*7425
74251
*74259
7426
*74265
*7427
74273
*74276
*74278
*74279
7428
*74283
*74290
*74293
*74298
*7430
*7432
*7433
*74351
74365
*74366
74367
*74368
7437
*74376
*74379
7438
*74390
*74393
7440
*7442
7445
*7446
7447
*7448
*7450
*7451
*7453
*7454
*7470
*7472
*7473
*7474
*7475
*7476
*7480
*7482
*7483
*7485
*7486
7488
*7489
*7490
*7491
*7492
*7493
*7494
*7495
*7496
*7497
74ALS00
74ALS01
74ALS02
74ALS03
74ALS04
74ALS05
74ALS08
74ALS09
74ALS10
74ALS109
*74ALS11
*74ALS112
74ALS114
74ALS12
*74ALS133
*74ALS15
74ALS20
*74ALS21
74ALS22
74ALS27
74ALS28
74ALS30
74ALS32
74ALS33
74ALS37
74ALS38
74ALS40
74ALS74
*74AS805
*74AS808
*74AS832
74C00
74C02
74C04
74C08
74C10
74C107
74C14
74C154
74C160
74C161
74C162
74C163
74C164
74C165
74C173
74C174
74C175
74C192
74C193
74C195
74C20
74C30
74C32
74C42
74C73
*74C74
74C76
74C83
*74C85
*74C86
74C90
*74C901
*74C902
74C903
74C904
*74C906
*74C907
*74C908
74C914
*74C918
*74C93
74C95
74F00
74F02
74F04
74F08
74F10
74F11
74F20
74F32
74F64
74F138
74F139
74F151
74F153
74F157
74F158
74F175
74F190
74F191
74F194
74F240
74F241
74F243
74F244
74F253
74F257
74F258
74F283
74F352
74F353
74F379
*74F533
*74F534
74H00
74H04
74H05
74H10
*74H103
*74H106
74H11
74H15
*74H183
74H20
74H21
74H22
74H30
74H40
74H50
74H51
*74H52
*74H53
*74H55
*74H71
74H72
74H73
74H74
74H76
*74H87
74L00
74L02
74L03
74L04
74L10
74L121
74L122
74L123
74L13
74L153
74L154
74L157
74L164
74L187A
74L192
74L193
74L20
74L30
74L42
74L422
74L46
74L47
*74L51
74L72
74L73
74L74
74L75
*74L78
74L85
74L86
74L90
74L91
74L93
74L96
*74L98
74LS00
74LS01
74LS02
74LS03
74LS04
74LS05
74LS08
74LS09
74LS10
74LS107
74LS109
74LS11
74LS112
*74LS113
74LS114
74LS12
74LS122
74LS123
74LS125
74LS126
74LS132
74LS136
74LS138
74LS139
74LS14
74LS145
74LS147
74LS148
74LS15
74LS151
74LS153
74LS155
74LS156
74LS157
74LS158
74LS160
74LS161
74LS162
74LS163
74LS163A
74LS164
74LS165
74LS166
74LS168
74LS169
74LS170
74LS173
74LS174
74LS175
74LS181
74LS183
74LS190
74LS191
74LS192
74LS193
74LS194
74LS195
74LS196
74LS197
74LS20
74LS21
74LS22
74LS221
74LS240
74LS241
74LS242
74LS243
74LS244
*74LS245
74LS251
74LS253
74LS257
74LS258
74LS259
74LS26
*74LS266
74LS27
74LS273
74LS279
74LS28
*74LS280
74LS283
74LS290
74LS293
74LS298
74LS299
74LS30
74LS32
74LS33
74LS347
*74LS348
*74LS352
*74LS353
74LS365
74LS366
74LS367
74LS368
74LS37
74LS373
74LS374
*74LS375
74LS377
74LS378
74LS379
74LS38
74LS390
74LS393
*74LS395
*74LS398
*74LS399
74LS40
74LS42
74LS445
74LS47
74LS48
74LS51
74LS533
74LS534
*74LS54
*74LS640
*74LS641
*74LS642
*74LS643
*74LS644
*74LS745
*74LS670
74LS73
74LS74
74LS75
74LS76
74LS78
74LS795
74LS796
74LS797
74LS798
74LS83
74LS85
74LS86
74LS90
74LS91
74LS92
74LS93
74LS95
74LS96
74S00
74S02
74S03
74S04
74S05
74S08
74S09
74S10
74S11
74S112
74S113
74S114
74S132
74S133
*74S134
*74S135
74S138
74S139
74S140
74S15
74S151
74S153
74S157
74S158
74S163
74S168
74S169
74S174
74S175
74S181
74S182
74S188
74S189
74S194
74S195
74S196
74S197
74S20
74S22
74S240
74S241
74S251
74S253

Table A-1. (Continued)

74S257
74S258
*74S260
*74S270
74S280
74S283
74S287
74S288
74S289
74S299
74S30
74S32
74S350
74S37
74S370
74S373
74S374
74S38
74S387
74S40
74S470
74S471
74S472
74S473
74S51
74S570
74S571
74S572
74S573
*74S64
*74S65
74S74
74S85
74S86
*75107
75107A
*75109A
*75113
*75114
*75115
*75116
*75117
*75118
*75119
*75121
*75122
75123
75124
*75125
*75127
*75128
*75129
*75136
*75138
*75140
*75142
*75150
*75152
*75154
*75158
*75159
*75182
*75183
75188
75189
75207
*75451
*75452
7573
7574
7577
7578
7597
7598
75S28
75S29
75S97
76L97
*81LS95
*81LS96
*81LS97
*81LS98
*8216
*8226
*8228
*82S100
82S123
82S126
82S129
*82S130
82S131
82S136
82S137
*82S184
*82S185
82S226
82S229
82S23
82S230
82S231
82S27
*82S62
*8531
*8T26
*8T28
8T95
8T96
8T97
8T98
9112
9114
9124
9147
9321
9334
93406
93417
93427
93434
93436
93438
93446
*93450
93451
93452
93453
93459
93488
*9401
*9601
*9602
AM25S08
AM25S09
*AM25S10
*AM25S18
*AM26LS31
*AM26LS32
AM26S10
*AM26S11
*AM2901A
AM2902
*AM2905
*AM2906
*AM2907
AM2908
*AM2909
*AM2910
*AM2911
*AM2913
*AM2914
AM2915A
AM2916A
*AM2918
*AM2919
*AM2921
*AM2932
*AM2940
*AM2942
*AM2950
AM29LS18
AM9016
*AM9080
AM9112
»*AM9513
*AM9519
*AMZ8001
*AMZ8002
*CD4000
*CD4001
*CD4002
*CD4006
*CD4008
*CD4009
*CD4010
*CD40100
CD40101
*CD40102
*CD40103
*CD40104
CD40106
*CD40107E
*CD40107F
*CD40109
*CD4011
*CD40110
*CD40114
*CD40115
*CD4012
*CD4013
*CD4014
*CD40147
*CD4015
CD40160
CD40161
CD40162
CD40163
*CD4017
CD40174
CD40175
*CD4018
CD40181
CD40182
*CD4019
CD40192
CD40193
CD40194
CD40195
*CD4020
*CD4021
*CD4022
*CD4023
*CD4024
*CD4025
CD40257
*CD4026
*CD4027
*CD4028
*CD4029
CD4030
*CD4031
*CD4032
*CD4033
*CD4034
*CD4035
*CD4037
*CD4038
*CD4040
*CD4041
*CD4042
*CD4043
*CD4044
*CD4048
CD4049
CD4050
*CD4054
*CD4055
*CD4056
*CD4059
*CD4060
*CD4063
*CD4068
CD4069
CD4070
*CD4071
*CD4072
*CD4073
*CD4075
*CD4076
*CD4077
*CD4078
*CD4081
*CD4082
*CD4085
*CD4086
*CD4089
*CD4093
*CD4094
*CD4095
*CD4096
*CD4098
*CD4099B
*CD4502
*CD4503
*CD4508B
*CD4511
*CD4512
*CD4514
*CD4515
*CD4516
*CD4518
*CD4519
*CD4520
*CD4527
*CD4532
*CD4543
*CD4553
*CD4555
*CD4556
*CD4557
*DIC14
*DIC16
*DIC18
*DIC20
*DIC22
*DIC24
*DIC26
*DIC28
*DIC30
*DIC32
*DIC34
*DIC36
*DIC38
*DIC40
*DIC8
DM8096
DM8097
»* DP8409
* DS7831
DS7832
* DS7833
DS8833
*F100101
*F100102
*F100107
*F100112
*F100114
*F100117
*F100118
*F100122
*F100130
*F100150
*F100151
*F100155
*F100156
*F100158
*F100163
*F100164
*F100165
*F100166
*F100170
*F100171
*F100179
*F100181
F16K
F34001
F34002
F34011
F34012
F34013
F34014
F34015
F340160
F340161
F340162
F340163
F34017
F340174
F340175
F34019
F340192
F340193
F340194
F340195
F34020
F34021
F34023
F34024
F34025
F34027
F34028
F34029
F34035
F34040
F34042
F34049
F34050
*F34068
F34071
F34077
F34078
F34081
F34085
F34086
F34518
F34520
F34555
FM4027
»*HM4864
»*HM50256
*HM6116
*HM6561
*HM6611
HM7602
HM7603
HM7608
HM7610
HM7611
HM7620
HM7621
HM7640
HM7641
HM7642
HM7643
HM7644
HM7648
HM7649
HM7680
*HM7680P
HM7681
*HM7681P
*HM7683
*HM7684
HM7684P
*HM7685
HM7685P
*HM7686
HM7686P
*HM7687
HM7687P
*I3232
*I3242
*I8031
*I8035
I8039
*I8048
I8049
*I8051
I8080
*I8085
»*I8086
»*I8087
»*I8088
»*I8089
*I8155
*I8156
*I8185
*I8202A
I8205
*I8212
*I8214
*I8218
*I8219
*I8224
*I8228
*I8237
»*I8251A
»*I8253
»*I8254
*I8255A
*I8257
*I8259A
*I8271
*I8272
*I8275
*I8279
*I8282
*I8283
*I8284
*I8284A
*I8286
*I8287
*I8288
*I8289
*I8355
I8748
I8749
I8751
I8755
*I8835
IM7027
IM7116
IM7141
*IMS1420
IMS1421
MB8168
*MC10100
*MC10101
*MC10102
*MC10103
*MC10104
*MC10105
*MC10106
*MC10107
*MC10108
*MC10109
*MC10110
*MC10111
*MC10113
*MC10114
*MC10115
*MC10116
*MC10117
*MC10118
*MC10119
*MC10121
*MC10123
*MC10124
*MC10125
*MC10128
*MC10129
*MC10130
*MC10131
*MC10132
*MC10133
*MC10134
*MC10135
*MC10136
*MC10137
*MC10138
*MC10141
*MC10153
*MC10158
*MC10159
*MC10160
*MC10161
*MC10162
*MC10163
*MC10164
*MC10165
*MC10166
*MC10168
*MC10170
*MC10171
*MC10172
*MC10173
*MC10174
*MC10175
*MC10176
*MC10177
*MC10178
*MC10179
*MC10180
*MC10181
*MC10182
*MC10183
*MC10186
*MC10188
*MC10189
*MC10190
*MC10191
*MC10193
*MC10194
*MC10195
*MC10197
MC10210
MC10211
*MC10212
MC10216
MC10231
*MC10287
MC14000
MC14001
MC14002
MC14006
MC14008
MC14011
MC14012
MC14013
MC14014
MC14015
MC14017
MC14018
MC14020
MC14021
MC14022
MC14023
MC14024
MC14025
MC14027
MC14028
MC14029
MC14032
MC14034
MC14035
MC14038
MC14040
MC14042
MC14043
MC14044
MC14049
MC14050
MC14068
MC14069
MC14070
MC14071
MC14072
MC14073
MC14075
MC14076
MC14077
MC14078
MC14081
MC14082
MC14086
MC14093
MC14094
MC14160
MC14161
MC14162
MC14163
MC14174
MC14175
MC14194
*MC14501
MC14502
MC14503
*MC14506
MC14507
MC14511
MC14512
MC14514
MC14515
MC14516
*MC14517
MC14518
MC14519
MC14520
*MC14522B
*MC14526B
MC14527
*MC14530
*MC14531B
MC14532
*MC14534
MC14543
MC14553
*MC14554
MC14555
MC14556
MC14557
*MC14558
*MC14560
*MC14561
*MC14562
*MC14566
*MC14569
*MC14572
*MC14580
MC14584
MC14585
*MC14599
*MC3242A
*MC3441
*MC3446A
*MC3447
*MC3448A
MC3459
*MC3486
*MC3487
»*MC6800
»MC68000
*MC6802
*MC6809
*MC6809E
*MC6820
*MC6821
MC6828
*MC6840
*MC6845
»*MC68451
*MC6850
»*MC68681
MC6880
MC6885
MC6886
MC6887
MC6888
MC68A09
MC68A09E
MC68B09
MC68B09E
*MCM10144
*MCM10145
*MCM10147
*MCM14505
*MCM14537
*MCM14552
MCM4027
MCM4116A
MCM6641
*MCM6665
*MCS6502
*MCS6503
*MCS6504
*MCS6505
*MCS6506
*MCS6512
*MCS6513
*MCS6514
*MCS6515
MCS6520
*MK36000
»*MK3880
*MK3881
*MK3882
*MK3884
MK4027
*MK4104
MK4114
MK4116
*MK4118
MK4802
MK68000
MM4601
MM4609
MM4611
MM4613
MM4615
MM4617
MM4619
MM4621
MM4623
MM4625
MM4627
MM4629
MM4635
MM4649
MM5257
MM5290
NE555
*PAL10H8
PAL10L8
*PAL12H6
PAL12L6
*PAL14H4
PAL14L4
*PAL16C1
*PAL16H2
PAL16L2
R6502
R6503
R6504
R6505
R6506
R6512
R6513
R6514
R6515
R6520
S4216
*S4264
S6809
S6809E
*S6834
S68A09
S68A09E
S68B09
S68B09E
»SC68681
SY6502
SY6503
SY6504
SY6505
SY6506
SY6512
SY6513
SY6514
SY6515
SY6520
SY6545
*TDC1008J
TMM2016
*TMS2532
TMS4027
TMS4044
TMS4045
*TMS4046
*TMS4047
TMS40L44
TMS40L45
TMS40L46
TMS40L47

Table A-1. (Continued)

TMS4116
*TMS4164
*TMS4416
*TMS9900
UPB403
UPB405
UPB406
UPB408
UPB425
UPB426
UPB427
UPB428
*UPD2364
UPD765
»Z80
Z8001
Z8002
»Z80CPU
Z80CTC
Z80PIO
Z80S10

Table A-2

List of Digital Devices in System Digital Library Sorted by Function.
(Prime types are listed only)

Function	Prime Type
4-BIT UNIVERSAL SHIFT REGISTER	74LS395
4-WIDE AND OR INVERT	74LS54
8192 X 8 READ ONLY MEMORY	UPD2364
8272 DUAL DENSITY FLOPPY DISK CONTROLLER	I8272
ACIA/UART	MC6850
ADDER,4-BIT	7483
ADDER,4-BIT	CD4008
ADDER,4-BIT	74283
ADDER,FULL,W/CARRY SAVE,DUAL	74H183
ADDER,FULL,BINARY,2-BIT	7482
ADDER,FULL,GATED	7480
ADDER,NBCD	MC14560
ADDER,SERIAL,TRIPLE	CD4038
ADDER,SERIAL,TRIPLE	CD4032
ADDER/SBTRCTR,2-BIT,DUAL	MC10180
ALU,4-BIT	MC10181
ALU,BINARY/BCD,4-BIT	F100181
ALU/FGRTR	25LS181
ALU/FGRTR,2-BIT	MC10182
ARBITER,BUS	I8289
BUFFER,9-BIT	F100122
BUFFER,HEX	CD4503
BUFFER,HEX	CD4010
BUFFER,QUAD	74126
BUFFER,QUAD	74125
BUFFER,TRUE/COMP,QUAD	CD4041
BUFFER,W/ENABLE,HEX	MC10188
BUFFER/DRVR,W/HI VOLT OUT,HEX	74C907
BUFFER/DRVR,W/HI VOLT OUT,HEX	74C906
BUFFER/DRVR,W/HI VOLT OUT,HEX	74C902
BUFFER/DRVR,2 IN NAND,DUAL	CD40107F
BUFFER/DRVR,2 IN NAND,DUAL	CD40107E
BUFFER/DRVR,W/HI VOLT OUT,HEX	7407
BUFFER/GATE,4 IN,DUAL	CD4012
BUFFER/INVRTR,STBD,HEX	CD4502
CHKR/GENRTR,CRC	9401
CMPLMNTR,9'S	MC14561
CMPRTR,4-BIT,MAGNITUDE	CD4063
CMPRTR,4-BIT,MAGNITUDE	74C85
CMPRTR,4-BIT,MAGNITUDE	7485
CMPRTR,5-BIT,MAGNITUDE	MC10166
CMPRTR,9-BIT	F100166
CNTLR, FLOPPY	I8271
CNTLR,BUS	I8288
CNTLR,BUS	I8218
CNTLR,CRT	I8275
CNTLR,CRT	MC6845
CNTLR,DMA,PGMBL	I8257
CNTLR,DYNAMIC RAM	DP8409
CNTLR,DYNAMIC RAM	I8202A
CNTLR,MICROPROGRAM	AM2910
CNTR,12-STAGE,RIPPLE	CD4040
CNTR,14-STAGE,RIPPLE	CD4020
CNTR,4-STAGE,U/D	MC10137
CNTR,4-STAGE,U/D	MC10136
CNTR,5 DECADE,REAL TIME	MC14534
CNTR,7-STAGE,RIPPLE	CD4024
CNTR,8-STAGE,DOWN,PRST SYNC	CD40103
CNTR,8-STAGE,DOWN,PRST SYNC	CD40102
CNTR,BCD,3-DIGIT	CD4553
CNTR,BCD,DUAL	CD4520
CNTR,BCD,DUAL	CD4518
CNTR,BCD,SYNC,U/D,W/CLR,DUAL	25LS192
CNTR,BCD,SYNC,U/D	25LS190
CNTR,BCD,U/D	CD4516
CNTR,BCD/BINARY,PGMBL,DUAL	MC14569
CNTR,BCD/DIVN/4-BIT,PGMBL	MC14522B
CNTR,BIN/DECADE,U/D,PRST	CD4029
CNTR,BIN/DIVN/4-BIT,PGMBL	MC14526B
CNTR,BINARY,SYNC,U/D,W/CLR,DUAL	25LS193
CNTR,BINARY,SYNC,4-BIT,U/D	25LS169A
CNTR,BINARY,SYNC,4-BIT,W/CLR	25LS163A
CNTR,BINARY,SYNC,4-BIT,W/DIRECT CLR	25LS161A
CNTR,BINARY,14-STAGE,RIPPLE-CARRY	CD4060
CNTR,BINARY,SYNC,U/D	74191
CNTR,BINARY	MC10178
CNTR,BINARY,4-BIT,DUAL	74393
CNTR,BINARY,4-BIT	74293
CNTR,BIQUINARY	MC10138
CNTR,DECADE,SYNC,4-BIT,U/D	25LS168A
CNTR,DECADE,SYNC,4-BIT,W/CLR	25LS162A
CNTR,DECADE,SYNC,4-BIT,W/DIRECT CLR	25LS160A
CNTR,DECADE,DIV2,DIV 5	7490
CNTR,DECADE,DUAL	74390
CNTR,DECADE,DIV2,DIV5	74290
CNTR,DIV12	7492
CNTR,DIV2,DIV8	74C93
CNTR,DIV2,DIV8	7493

Function	Prime Type
CNTR,DIVN,PGMBL	CD4059
CNTR,DIVN,PRST	CD4018
CNTR/DCDR,DEC,W/LATCH DRVR,U/D,COS/MOS	CD40110
CNTR/DVDR,DEC	CD4033
CNTR/DVDR,DEC	CD4026
CNTR/DVDR,OCTAL	CD4022
CNTR/DVDR,DECIMAL	CD4017
CNTR/LATCH,BINARY,PRST	74177
CNTR/LATCH,DECADE/BIQUINARY,PRST	74176
CNVRTR,BCD TO BINARY	74184
CNVRTR,BINARY TO BCD	74185
CNVRTR,LEVEL,BIDI,TTL INTERFACE,8-BIT	CD40115
CNVRTR,LEVEL,QUAD	CD40109
CONTROLLER,BUS,UCOMPUTER,BIPOLAR	I8219
CPU,16-BIT,HMOS	68000
CPU,4-BIT,SLICE,BIPOLAR MICRO	AM2901A
CPU,8-BIT	I8085
CPU,8-BIT	MC6800
CPU,8-BIT	AM9080
CPU,8-BIT	MK3880
CPU,8-BIT,W/RAM+TIMER	MC6802
CPU,8-BIT.MOS TECH.	MCS6502
CPU,MOS TECH.	MCS6514
CPU,MOS TECH.	MCS6515
CPU,MOS TECH.	MCS6513
CPU,MOS TECH.	MCS6512
CPU,MOS TECH.	MCS6506
CPU,MOS TECH.	MCS6505
CPU,MOS TECH.	MCS6504
CPU,MOS TECH.	MCS6503
CTC/TIMER	MK3882
DCDR,1-OF-8	AM2921
DCDR,1-OF-8	MC10162
DCDR,1-OF-8	MC10161
DCDR,BCD TO DECIMAL	CD4028
DCDR,BCD TO 7-SEG	MC14558
DCDR,BCD TO 7-SEG	7448
DCDR,BCD TO 7-SEG	7446
DCDR,BCD TO DEC	7442
DCDR/DEMUX,4-TO-16	74159
DCDR/DEMUX	74156
DCDR/DEMUX	74155
DCDR/DEMUX,4-TO-16	74154
DCDR/DEMUX,1-OF-4,DUAL	MC10172
DCDR/DEMUX,1-OF-4,DUAL	MC10171
DCDR/DEMUX,UNIV	F100170
DCDR/DRVR,W/LATCH,BCD TO 7-SEG	CD4056
DCDR/DRVR,BCD TO 7-SEG	CD4055
DCDR/DRVR,BCD TO DECIMAL	74145
DCDR/MUX,2-TO-4,DUAL	25LS139
DCDR/MUX,2-TO-4 LINE,DUAL	CD4556
DCDR/MUX,2-TO-4 LINE,DUAL	CD4555
DCDR/MUX,3-TO-8	25LS138
DET/COR,ERROR	MC10163
DET/COR,ERROR	MC10193
DMA	I8237
DRIVER,QUAD	F100112
DRVR,30 VOLT,DUAL	74C918
DRVR,30 VOLT,DUAL	74C908
DRVR,AND,DUAL	75451
DRVR,BUS,4-BIT,BIDI,PARALLEL	8226
DRVR,BUS,4-BIT,BIDI,PARALLEL	8216
DRVR,BUS,BIDI,PARALLEL,4-BIT	DS7833
DRVR,BUS,HEX	70C95
DRVR,BUS,HEX	70C97
DRVR,BUS,HEX	74368
DRVR,BUS,HEX	74366
DRVR,BUS,MECL TO TTL/IBM,DUAL	MC10128
DRVR.BUS.OCTAL	I81LS97
DRVR,BUS,OCTAL	81LS95
DRVR,BUS,OCTAL,INV	81LS96
DRVR,BUS,OCTAL,INV	81LS98
DRVR,BUS,QUAD	75138
DRVR,BUS,QUAD	AM26S11
DRVR,BUS,TRIPLE	MC10123
DRVR,DISPLAY,4-SEG	CD4054
DRVR,LINE	DS7831
DRVR,LINE,DIFF,DUAL	75183
DRVR,LINE,DIFF,DUAL	75159
DRVR,LINE,DIFF,DUAL	75158
DRVR,LINE,DUAL	75150
DRVR,LINE,QUAD	MC3487
DRVR,LINE,TRIPLE	75122
DRVR,LINE,DUAL	75121
DRVR,LINE,DIFF,DUAL	75114
DRVR,LINE,DIFF,DUAL	75113
DRVR,LINE,DIFF,QUAD,RS-422	AM26LS31
DRVR,LINE,DUAL	75109A
DRVR,LINE,QUAD	1488

Table A-2. (Continued)

Function	Prime Type
DRVR,NAND	75452
DRVR/CNTLR,SYSTEM AND BUS	I8228
DRVR/GENRTR,CLK	I8224
DRVR/PLSCZR,DUAL	74120
DUAL PARITY CHECKER-GENERATOR	100160
DUMMY IC	DIC8
DUMMY IC	DIC40
DUMMY IC	DIC38
DUMMY IC	DIC36
DUMMY IC	DIC34
DUMMY IC	DIC32
DUMMY IC	DIC30
DUMMY IC	DIC28
DUMMY IC	DIC26
DUMMY IC	DIC24
DUMMY IC	DIC22
DUMMY IC	DIC20
DUMMY IC	DIC18
DUMMY IC	DIC16
DUMMY IC	DIC14
ECL RAM TEST 1K X 1 STATIC	10415
ELMNTS,4-BIT,TRUE/COMP,ONE/ZERO	74H87
ELMNTS,COMP OUT,QUAD	74265
ENCDR,PRIO,8-LINE TO 3-LINE	74LS348
ENCDR,PRIO,UNIV	F100165
ENCDR,PRIO	CD4532
ENCDR,PRIO,8-LINE TO 3-LINE	74148
ENCDR,PRIO,10-LINE DECIMAL TO 4-LINE BCD	74147
EPROM,1K × 8	2708
EPROM,2K × 8	2716
EPROM,4K × 8	2732
EPROM,8K × 8	2764
EPROM16K × 8	27128
FF,D	CD4013
FF,D,HEX	25LS174
FF,D,HEX	25LS07
FF,D,HIGH SPEED,HEX	MC10176
FF,D,MS,DUAL	MC10131
FF,D,OCTAL	25LS377
FF,D,OCTAL	25LS374
FF,D,OCTAL	25LS273
FF,OCTAL,3-STATE	74F534
FF,D,QUAD	25LS175
FF,D,QUAD	CD4076
FF,D,QUAD	74379
FF,D,W/PRE+CLR,DUAL	74C74
FF,D,W/PRE+CLR,DUAL	7474
FF,D,W/RESET,HEX	MC10186
FF,HEX D	F100151
FF,JK	7472
FF,JK	7470
FF,JK,DUAL	74H106
FF,JK,DUAL	7476
FF,JK,DUAL	7473
FF,JK,L H L PULSE,DUAL	74107
FF,JK,MS,DUAL	MC10135
FF,JK,MS,W/SET+RESET,DUAL	CD4027
FF,JK,QUAD	74376
FF,JK,QUAD	74276
FF,JK,W/3 IN AND	CD4095
FF,JK,W/3 IN AND	CD4096
FF,JK,W/3 IN AND+DATA LOCKOUT	74110
FF,JK,W/CLR,DUAL	74H103
FF,JK,W/DATA LOCKOUT,DUAL	74111
FF,JK,W/PRE+CLR,DUAL	74ALS112
FF,JK,W/PRE+COMMON CLR+COMMON CLK,DUAL	74L78
FF,JK,W/PRE+CLK+CLR,DUAL	74114
FF,JK,W/PRE+CLR,DUAL	74109
FF,JK,W/PRE,DUAL	74LS113
FF,RS,MS,W/PRE+CLR,AND-GATED	74H71
FPLA	82S100
GATE,4-3-3-3 IN,4-WIDE	MC10121
GATE,AND,2 IN,QUAD	CD4081
GATE,AND,2 IN,QUAD	MC10104
GATE,AND,2 IN,QUAD	7409
GATE,AND,2 IN,QUAD	7408
GATE,AND,3X3	CD4073
GATE,AND,3 IN,TRIPLE	74ALS15
GATE,AND,3 IN,TRIPLE	74ALS11
GATE,AND,4 IN,DUAL	CD4082
GATE,AND,4 IN,DUAL	74ALS21
GATE,AND,HEX	MC10197
GATE,AND-NAND,8 IN	CD4068
GATE,AND-OR-INV,EXP	CD4086
GATE,AND-OR-INV,2 IN,2-WIDE,DUAL	CD4085
GATE,AND-OR,BI-PHASE PAIRS,TRIPLE	CD4037
GATE,AND-OR-INV,EXP 2 WIDE 4 IN	74H55
GATE,AND-OR-INV,EXP 4-WIDE,W/3 IN AND	74H53
GATE,AND-OR,EXP 4-WIDE	74H52
GATE,AND-OR-INV,2-3 IN,2-WIDE,OA/OA/INV,DUAL	MC10117
GATE,AND-OR-INV,2-WIDE OA/OAI,TRIPLE	F100117
GATE,AND-OR-INV	74L51
GATE,AND-OR-INV,4-2-3-2	74S65
GATE,AND-OR-INV,4-2-3-2	74S64
GATE,AND-OR,SEL,QUAD	CD4019
GATE,AND-OR-INV,4-WIDE	7454
GATE,AND-OR-INV,EXP,4-WIDE	7453
GATE,AND-OR-INV	7451
GATE,AND-OR-INV,2X2,DUAL	7450
GATE,AND-OR-INV,2 IN,2-WIDE EXPANDABALE,DUAL	MC14506
GATE,AND/NAND,4 IN,DUAL	MC10108
GATE,INV,HEX	74C901
GATE,INV,HEX	CD4009
GATE,INVRTR,HEX	7405
GATE,INVRTR,HEX	7404
GATE,MAJLGC,5 IN,DUAL	MC14530
GATE,MLTIF,EXP	CD4048
GATE,MLTIF,TRIPLE	MC14501
GATE,NAND,2 IN,SCHMIDT TRIG,QUAD	CD4093
GATE,NAND,13 IN	74ALS133
GATE,NAND,4 IN,DUAL	7422
GATE,NAND,3 IN,TRIPLE	CD4023
GATE,NAND,2 IN,QUAD	CD4011
GATE,NAND,3 IN,TRIPLE	7412
GATE,NAND,3 IN,TRIPLE	7410
GATE,NAND,2 IN,QUAD	7403
GATE,NAND,2 IN,QUAD	7401
GATE,NAND,2 IN,QUAD	7400
GATE,NAND,12 IN	74S134
GATE,NAND,8 IN	F34068
GATE,NAND,8 IN	7430
GATE,NAND/NOR/INV,4 INV,2 IN NOR,2 IN NAND,HEX	MC14572
GATE,NOR,2 IN,QUAD	MC10102
GATE,NOR,2 IN,W/STB,QUAD	MC10100
GATE,NOR,2X4 W/STB	CD4002
GATE,NOR,2 IN,QUAD	CD4001
GATE,NOR,2 IN,QUAD	7402
GATE,NOR,2X4,W/STB	7425
GATE,NOR,3 IN,TRIPLE	CD4075
GATE,NOR,3 IN 3 OUT,DUAL	MC10111
GATE,NOR,3 IN,TRIPLE	CD4025
GATE,NOR,3X3	7427
GATE,NOR,4-3-3 IN,TRIPLE	MC10106
GATE,NOR,4X2	7433
GATE,NOR,5 IN,DUAL	74S260
GATE,NOR,8 IN	CD4078
GATE,NOR,EXP,2X4,W/STB	7423
GATE,NOR/INV,3 IN NOR+INVRTR,DUAL	CD4000
GATE,OA/OAI	F100118
GATE,OR,2 IN,QUAD	MC10103
GATE,OR,3 IN 3 OUT,DUAL	MC10110
GATE,OR,4 IN,DUAL	CD4072
GATE,OR,4X2	CD4071
GATE,OR,4X2	7432
GATE,OR/AND,4-3-3-3 IN,4-WIDE	MC10119
GATE,OR/AND,3 IN,2-WIDE,DUAL	MC10118
GATE,OR/NOR,4-5 IN,DUAL	MC10109
GATE,OR/NOR,2-3-2 IN,TRIPLE	MC10105
GATE,OR/NOR,QUAD	MC10101
GATE,OR/NOR,2 IN,QUINT	F100102
GATE,OR/NOR,5 IN,TRIPLE	F100101
GATE,OR/NOR,3 IN 3 OUT,DUAL	MC10212
GATE,XOR,QUAD	CD4077
GATE,XOR,QUAD	74C86
GATE,XOR,QUAD	MC10113
GATE,XOR,QUAD	74LS266
GATE,XOR,QUAD	74136
GATE,XOR/XNOR,2 IN,TRIPLE	MC10107
GATE,XOR/XNOR,2 IN,QUINT	F100107
GATE,XOR/XNOR,QUAD	74S135
GATE/BUFFER,NAND,4 IN,DUAL	7413
GENRTR,ADDR,DMA	AM2940
GENRTR,CARRY LOOK AHEAD	F100179
GENRTR,CARRY,LOOK AHEAD	74182
GENRTR,CLK	I8284
GENRTR,DMA,CNTR/TIMER	AM2942
GENRTR,LOOK AHEAD CARRY	MC10179
GENRTR,TIMEBASE,INDUSTRIAL	MC14566
GENRTR/CHKR,PARITY,9-BIT	82S62
GENRTR/CHKR,PARITY,9-BIT	74LS280
GENRTR/CHKR,PARITY,9-BIT,ODD/EVEN	74180
GENRTR/CHKR,PARITY,9+2	MC10170
GENRTR/CHKR,PARITY,12-BIT	MC10160
HEX 2 INPUT OR DRIVER	74AS832
HEX 2 INPUT AND DRIVER	74AS808
HEX 2 INPUT NOR DRIVER	74AS805
I/O PROCESSOR	I8089
INTERRUPT CONTROLLER,PROG	AM9519
INVRTR,W/ENABLE,HEX	MC10189
INVRTR/BUFFER,HEX	MC10195

Table A-2. (Continued)

Function	Prime Type	Function	Prime Type
KEYBOARD INTERFACE,PROG	I8279	RAM,128 X 1,MECL	MCM10147
LATCH,4-BIT,DUAL	CD4508B	RAM,16 X 4	3101
LATCH,4-BIT,DISTABLE	74LS375	RAM,16 X 4	74189
LATCH,4-BIT,DUAL	74116	RAM,16 X 4	CD40114
LATCH,4-BIT	7475	RAM,16 X 4	7489
LATCH,8-BIT,ADDRESSABLE	CD4099B	RAM,16K X 4,DYNAMIC	TMS4416
LATCH,8-BIT,ADDRESSABLE	74259	RAM,16K X 1,W/MPXED ADDR	2117
LATCH,8-BIT	74100	RAM,1K X 1,STATIC	2125A
LATCH,D,CLKED,QUAD	CD4042	RAM,1K X 1,STATIC	2115A
LATCH,D,OCTAL	25LS373	RAM,1K X 1,STATIC	2102A
LATCH,D,TRIPLE	F100130	RAM,1K X 4,STATIC	TMS4047
LATCH,DUAL	MC10130	RAM,1K X 4,STATIC	2142
LATCH,HEX D	F100150	RAM,1K X 4,STATIC	2114
LATCH,OCTAL	I8283	RAM,1K X 8	I8185
LATCH,OCTAL	I8282	RAM,1K X 8,STATIC	MK4118
LATCH,OCTAL,3-STATE	74F533	RAM,256 X 1,MECL	MCM10144
LATCH,QUAD	MC10153	RAM,256 X 1	74200
LATCH,QUAD	MC10133	RAM,256 X 1,STATIC CMOS	MCM14537
LATCH,QUAD	MC10168	RAM,256 X 1,STATIC CMOS	MCM14552
LATCH,QUINT	MC10175	RAM,256 X 4,STATIC	2112A
LATCH,RS,NAND,QUAD	CD4044	RAM,256 X 4,STATIC	2111A
LATCH,RS,NOR,QUAD	CD4043	RAM,256 X 4,STATIC	2101A
LATCH,RS,QUAD	74279	RAM,256 X 4	HM6561
LATCH/DCDR/DRVR,BCD TO 7-SEG	CD4543	RAM,256K X 1,DYNAMIC	HM50256
LATCH/DCDR,4-BIT LATCH,1-T0-16 DCDR	CD4515	RAM,2K X 8,STATIC	MK4802
LATCH/DCDR,4-BIT LATCH,1-T0-16 DCDR	CD4514	RAM,4K X 1,STATIC	TMS4046
LATCH/DCDR/DRVR,BCD TO 7-SEG	CD4511	RAM,4K X 1,STATIC	MK4104
LATCH/PORT,8-BIT,BIDI,ADDRESSABLE	MC14599	RAM,4K X 1,STATIC	2147
MASK/MERGE	F100156	RAM,4K X 1,DYNAMIC	2104A
MATRIX,SHIFT,8-IN	F100158	RAM,4K X 1,STATIC	145101
MICROPROCESSOR	TMS9900	RAM,4K X 4,DYNAMIC	IMS1420
MLTPLR,2X1	MC10287	RAM,64 X 1,STATIC	MCM14505
MLTPLR,4X2	MC10183	RAM,64K X 1	TMS4164
MLTPLR,6-BIT,RATE	7497	RAM,64K X 1	HM4864
MLTPLR,BCD,RATE	CD4527	RCVR,BUS,TTL/IBM TO MECL,QUAD	MC10129
MLTPLR,BINARY,RATE	CD4089	RCVR,LINE,DIFF,DUAL	75182
MLTPLR,BINARY,2X2,PARALLEL	MC14554	RCVR,LINE,DIFF,QUAD,RS-422	AM26LS32
MLTPLR/ACCUM,TRW	TDC1008J	RCVR,LINE,QUAD	75154
MLTVB,MONO,DUAL	CD4098	RCVR,LINE,DUAL	75152
MLTVB,MONO	74121	RCVR,LINE,TRIPLE	MC10116
MLTVB,RETRIG MONO,DUAL	74221	RCVR,LINE,QUAD	MC10115
MLTVB,RETRIG MONO,DUAL	74123	RCVR,LINE,TRIPLE	MC10114
MLTVB,RETRIG MONO	74122	RCVR,LINE,QUAD	MC3486
MEMORY MANAGEMENT UNIT	MC68451	RCVR,LINE,DIFF,QUINT	F100114
MPU	MC6809E	RCVR,LINE,DUAL	75142
MPU	MC6809	RCVR,LINE,DUAL	75140
MPU,16-BIT	I8088	RCVR,LINE,8-CHANNEL	75129
MPU,16-BIT	I8086	RCVR,LINE,8-CHANNEL	75128
MPU,8-BIT	I8035	RCVR,LINE,7-CHANNEL	75127
MPU,8-BIT	I8046	RCVR,LINE,7-CHANNEL	75125
MPU,8-BIT	I8051	RCVR,LINE,DIFF,DUAL	75115
MPU,8-BIT,ROMLESS	I8031	RCVR,LINE,DUAL	75107
MUX,16-IN	F100164	RCVR,LINE,QUAD	1489
MUX,2 IN,QUAD	MC10159	REG,4-BIT,STATIC,DUAL	CD4015
MUX,2 IN,TRUE OUT,QUAD	MC10158	REG,D,4-BIT	74173
MUX,2 IN,W/STRG,QUAD	74LS399	REG,D,W/STD AND OUT,QUAD	AM2918
MUX,2 IN,W/STRG,QUAD	74LS398	REG,FILE,16X4	MCM10145
MUX,4 IN,W/ENABLE,TRIPLE	F100171	REG,FILE,16-BIT	74172
MUX,8 TO 1, DUAL	74351	REG,FILE,4X4	74LS670
MUX,8-IN,DUAL	F100163	REG,FILE,4X4	74170
MUX,8-LINE	MC10164	REG,MULTIPORT,4X4	MC14580
MUX.DUAL	MC10174	REG,PRIO,4-BIT,CASCADEABLE	74278
MUX,MEM ADDR+REFRESH CNTR	MC3242A	REG,QUAD	AM2919
MUX,W/LATCH,DUAL	MC10134	REG,QUAD D	AM25S18
MUX,W/LATCH+COMMON RESET,DUAL	MC10132	REG/SEL,STRG,4-BIT,DATA	74L98
MUX/CNTR,INTEL DYNAMIC RAM RFSH	I3242	ROM,1K X 4	8228
MUX/CNTR,INTEL DYNAMIC RAM RFSH	I3232	ROM,1K X 4	3625
MUX/LATCH,2-IN,QUAD	F100155	ROM,1K X 4	3605
MUX/LATCH,QUAD	MC10173	ROM,1K X 4	6253
NONSEGMENTED 16-BIT CPU	AMZ8002	ROM,1K X 4	6252
ONE-SHOT	9601	ROM,1K X 4	6251
ONE-SHOT,DUAL	9602	ROM,1K X 4	6250
PIA/PORT	MC6820	ROM,1K X 8	2608
PIC	AM2914	ROM,1K X 8	4700
PIC,EXPANDER	AM2913	ROM,1K X 8	3628
PICU	I8259A	ROM,1K X 8	3608
PICU	I8214	ROM,1K X 8	HM7683
PICU	MC6828	ROM,1K X 8	93450
PIO/PORT	MK3881	ROM,1K X 8	6289
PLA	PAL16H2	ROM,1K X 8	6287
PLA	PAL16C1	ROM,1K X 8	6286
PLA	PAL14H4	ROM,1K X 8	6284
PLA	PAL12H6	ROM,1K X 8	2308
PLA	PAL10H8	ROM,1K X 8	HM7681P
PORT,8-BIT,I/O	I8212	ROM,1K X 8	HM7680P
PORT,8-BIT,BIDI,IO	AM2950	ROM,256 X 8	29601
PORT,INTEL PGMBL PERIPH. INTERFACE	I8255A	ROM,256 X 8	29600
PORT/TIMER,W/RAM 2K X 8	I8156	ROM,256 X 8	6236
PORT/TIMER,W/RAM 2K X 8	I8155	ROM,256 X 8	6235
PTM/TIMER	MC6840	ROM,256 X 4	4210
RAM TEST 4K X 4 STATIC	2168	ROM,256 X 4	29661

Table A-2. (Continued)

Function	Prime Type	Function	Prime Type
ROM,256 X 4	29660	SHFTRG,4-BIT	7494
ROM,256 X 8	1702	SHFTRG,5-BIT	7496
ROM,256 X 4	HM6611	SHFTRG,64-STAGE	CD4031
ROM,2K X 4	82S185	SHFTRG,64-BIT,STATIC,DUAL	MC14517
ROM,2K X 4	82S184	SHFTRG,8-BIT,BIDI,UNIV,W/STRG	25LS299
ROM,2K X 8	8531	SHFTRG,8-BIT,PARALLEL/SERIAL	CD4014
ROM,2K X 8	6275	SHFTRG,8-BIT,BIDI	74199
ROM,2K X 8	4246	SHFTRG,8-BIT,BIDI,UNIV	74198
ROM,2K X 8	HM7687	SHFTRG,8-BIT,BIDI,SERIAL/PARALLEL	CD4034
ROM,2K X 8	HM7686	SHFTRG,8-BIT	74166
ROM,2K X 8	HM7685	SHFTRG,8-BIT,W/PARALLEL LOAD+INV OUT	74165
ROM,2K X 8	2316E	SHFTRG,8-BIT,SI/PO	25LS164
ROM,2K X 8	HM7684	SHFTRG,8-BIT	7491
ROM,32 X 8	6231	SHFTRG,8-STAGE,STATIC	CD4021
ROM,32 X 8	6230	SHFTRG,UNIV	MC10141
ROM,4K X 8	TMS2532	SHFTRG/STRG,8-STAGE BUS	CD4094
ROM,4K X 8	2332	SIO/USART	MK3884
ROM,512 X 4	82S130	SQNCR,MICROPROGRAM	AM2911
ROM,512 X 8	29625	SQNCR,MICROPROGRAM	AM2909
ROM,512 X 8	29624	STACK,PUSH/POP	AM2932
ROM,512 X 8	29623	TIMER/CNTR,INTEL PGMBL CLK	I8253
ROM,512 X 8	29622	TIMER,PROG,INTERVAL	I8254
ROM,512 X 8	29621	TIMER/ONE-SHOT	555
ROM,512 X 8	29620	TIMING CNTLR,SYSTEM	AM9513
ROM,512 X 4	29611	TREE,PARITY,12-BIT	MC14531B
ROM,512 X 4	29610	TRNSLTR,MECL TO TTL,QUAD	MC10125
ROM,512 X 8	S6834	TRNSLTR,MECL TO MST,HEX	MC10191
ROM,512 X 8	4204	TRNSLTR,MECL TO NMOS,TRIPLE	MC10177
ROM,512 X 4	74S270	TRNSLTR,MST TO MECL,QUAD	MC10190
ROM,8K X 8	S4264	TRNSLTR,TTL TO MECL,QUAD	MC10124
ROM,8K X 8	2364	UART,DUAL	MC68681
ROM,8K X 8	MK36000	USART	I8251A
ROM/PORT,2K X 8	I8355	XCVR,BUS,2 IN,W/PARITY,QUAD	AM2906
SEGMENTED 16-BIT CPU	AMZ8001	XCVR,BUS,2 IN,OC DRVR W/TRI RCVR,QUAD	AM2905
SEL,2-TO-1,QUAD	25LS157	XCVR,BUS,BIDIR	MC3448A
SEL,8-CHANNEL,DATA	CD4512	XCVR,BUS,OCTAL	I8287
SEL,AND/OR,4-BIT	CD4519	XCVR,BUS,OCTAL	I8286
SEL/MUX,1-OF-8	74152	XCVR,BUS,OCTAL	74LS245
SEL/MUX,1-OF-16	74150	XCVR,BUS,OCTAL,3-STATE	74LS645
SEL/MUX,1-OF-8	25LS151	XCVR,BUS,OCTAL,3-STATE,INV+TRUE	74LS643
SEL/MUX,2-TO-1,DUAL	25LS158	XCVR,BUS,OCTAL,3-STATE,INV	74LS640
SEL/MUX,4-TO-1,DUAL	25LS153	XCVR,BUS,OCTAL,GPIB	MC3447
SEL/MUX,4-TO-1,DUAL	74LS352	XCVR,BUS,OCTAL,O/C,INV+TRUE	74LS644
SEL/MUX,DATA,W/INV OUT,QUAD	25LS258	XCVR,BUS,OCTAL,O/C,INV	74LS642
SEL/MUX,DATA,W/NINV OUT,QUAD	25LS257	XCVR,BUS,OCTAL,O/C	74LS641
SEL/MUX,DATA,DUAL	25LS253	XCVR,BUS,QUAD	I8835
SEL/MUX,DATA	25LS251	XCVR,BUS,QUAD	75136
SEL/MUX,DATA,DUAL	74LS353	XCVR,BUS,QUAD	25LS243
SEL/STRG,4-BIT,DATA	74298	XCVR,BUS,QUAD	25LS242
SHIFTER,4-BIT,3-STATE	AM25S10	XCVR,BUS,QUAD,GPIB	MC3446A
SHFTRG,1 TO 64-BIT,VARIABLE	CD4557	XCVR,BUS,QUAD,GPIB	MC3441
SHFTRG,128-BIT,STATIC	MC14562	XCVR,BUS,SIMULT,DUAL	MC10194
SHFTRG,18-STAGE,STATIC	CD4006	XCVR,BUS,W/INTERFACE LOGIC,QUAD	AM2907
SHFTRG,200-STAGE,DYNMC,CMOS	4062A	XCVR,LINE,DIFF	75119
SHFTRG,32-STAGE,LEFT/RIGHT	CD40100	XCVR,LINE,DIFF	75118
SHFTRG,4 BIT,BIDI,UNIV	CD40104	XCVR,LINE,DIFF	75117
SHFTRG,4-BIT,PI/PO	CD4035	XCVR,LINE,DIFF	75116
SHFTRG,4-BIT,W/PARALLEL LOAD	25LS195A	XCVR/BUFFER,LINE,OCTAL	25LS244
SHFTRG,4-BIT,BIDI,UNIV	25LS194A	XCVR/BUFFER,LINE,OCTAL	25LS241
SHFTRG,4-BIT	7495	XCVR/BUFFER,LINE,OCTAL	25LS240

System Error Messages – Appendix B

B.1 INTRODUCTION

This section contains listings of errors commonly reported by the board test system. Para B.2 lists errors from the Circuit Description Compiler. Para B.3 contains errors generated by the Test Program Translator. Para B.4 provides a listing of Run Time Diagnostic errors. Para B.5 provides a list of errors generated by the text editor and the various RSX-11M System programs.

B.2 CIRCUIT DESCRIPTION COMPILATION ERRORS

B.2.1 Pass 1 Errors

ADAPTOR SECTION IS NOT ALLOWED IN COMPONENT DESCRIPTION

BAD CHARACTER IN STRING

CHARACTERISTIC VALUE MUST BE GIVEN

R, L, C must have a value, or if a tolerance is specified a value is needed.

CHARACTERISTIC VALUE PREVIOUSLY DEFINED

Tolerance value may be improperly entered and looks like a characteristic value.

COMMENT PAST END OF FILE

COMPUTED INVALID PIN FOR THIS DEVICE

Too many commas in the CONNECTION field.

DCR FLAGSPEC ON DEVICE (TYPE L) REQUIRED IN COMPONENT LIB

DEVICE MISSING FROM ANALOG COMPONENT LIBRARIES

DEVICE xx HAS NO VALUE SPECIFIED

DEVICE xx IS MISSING nn PIN(S) FOR PROPER CONNECTION

DEVICE HAS GREATER THAN nn LEVEL(S) OF NESTING

DEVICE HAS PREVIOUSLY BEEN ASSIGNED A VALUE

Device listed twice or same name used for two devices.

DEVICE NOT PREVIOUSLY DEFINED

DEVICE TYPE DIFFERENT THAN WHEN FIRST DEFINED AT LINE nn

Type specified in CIRCUIT section inconsistent with pin type or value spec.

DEVICE TYPE CANNOT BE A NAIL TYPE

DIGITAL IC PIN HAS AN INVALID NAME, xx, IN COMPONENT LIBRARY

EMPTY INPUT FILE	Wrong file device, name, or extension was probably indicated on option page.
ERROR LIMIT EXCEEDED	The compiler stops after 200 errors have been found.
FLAG SPECIFICATION MUST START WITH ALPHA CHARACTER	
HIGH TOLERANCE PREVIOUSLY DEFINED	Probably was intended to be the negative tolerance spec.
INVALID EXTERNAL NAME	A character other than a letter or digit was detected.
INVALID PIN FOR THIS DEVICE	Pin name not consistent with device type.
INVALID SYNTAX ITEM xx	
INVALID VALUE ITEM	
LINE IS TOO LONG AND HAS BEEN TRUNCATED	
LOW TOLERANCE PREVIOUSLY DEFINED	Probably was intended to be the positive tolerance spec.
MESSAGE TOO LONG	
MISSING CONNECTION LIST FOR THIS EXTERNAL	
MISSING OR BAD %CIRCUIT STATEMENT	
NAIL SPECIFIED IS NOT WITHIN ALLOWABLE RANGE	
NEW LINE INSIDE A QUOTE	
NO VALUE HAS BEEN SPECIFIED FOR DEVICE xx	
NODE NAME HAS MORE THAN ONE EXTERNAL AT LINE nn	
MULTIPLE PROBE CONNECTIONS SPECIFIED	The probe may only be connected to one adaptor pin.
PIN NAME MUST BE GIVEN EXPLICITLY	Alphabetic pin names must be given in CONNECTION field.
PREMATURE END OF FILE	
PROBE CONNECTION PREVIOUSLY DEFINED	The probe may only be connected to one adaptor pin.
PROBE MUST BE CONNECTED TO A DIGITAL DRIVER/SENSOR	The probe connection is indicated at an analog pin.
TESTER PIN NUMBER MUST BE GIVEN	Node was defined in CIRCUIT section but not in ADAPTOR section.
THIS DEVICE HAS BEEN PREVIOUSLY DEFINED	

THIS DEVICE TYPE IS INVALID IN A COMPONENT DESCRIPTION

THIS EXTERNAL HAS BEEN PREVIOUSLY DEFINED AT LINE nn

The same name is used for two external nodes.

THIS EXTERNAL IS ALREADY CONNECTED TO NODE xx AT LINE nn

THIS PIN IS ALREADY CONNECTED TO NODE xx AT LINE nn

THIS PIN NAME HAS BEEN PREVIOUSLY DEFINED

The same name is used for two external nodes.

THIS TESTER PIN NUMBER HAS BEEN PREVIOUSLY DEFINED AT LINE nn

This pin number has already been used. (For instance, pin number n could have been specified for both an analog pin and a digital pin. The same pin number cannot be used for two different functions.)

THIS TESTER PIN PREVIOUSLY CONNECTED TO NODE xx AT LINE nn

YOU HAVE A SYNTAX ERROR

The statement is improperly constructed. Refer to CDL definition for syntax rules.

B.2.2 Pass 1 Warnings

CHARACTERISTIC VALUE FOR THIS DEVICE IS OUT OF USUAL VALUE RANGE

See Table 7-3, in para 7.4, for a list of usual value ranges.

DEVICE NOT IN ANY LIBRARY, DEFAULTS TO nn PIN IC

Any time an unrecognizable CDL symbol is used for a device type, ATG assumes it is an IC type name and searches the digital libraries for a matching name. If one is not found, ATG assumes the device is an IC that is not in any library.

EXTRANEOUS <CR> FOUND AND REMOVED IN SOURCE TEXT

ATG does not respond to the carriage return code. New line (NL) code is used for CR/LF.

TARGET SYSTEM TYPE AND ADAPTER SECTION TYPE ARE DIFFERENT

TESTER PIN GIVEN BUT IGNORED

One possible reason for this message is that a comma may not have preceded the first connection in the connection field of a CIRCUIT EXT statement. ATG does not acknowledge such a connection; therefore, it is unknown to ATG during the processing of the adaptor section.

THIS DEVICE IS OF A DIFFERENT TYPE THAN THOSE PREVIOUSLY MENTIONED

TOO MANY CHARACTERS

UNSPECIFIED TESTER PIN TYPES DEFAULT TO TYPE ANALOG

B.2.3 Pass 2 Errors

INTERMEDIATE NAME NOT USED WHEN NODE FIRST DEFINED AT LINE nn	If an intermediate name is to be applied to a node it must be included in the first definition of the node.
THIS DEVICE OR PIN NAME HAS NOT BEEN DEFINED ANYWHERE	Reference is made to a device or a device pin which was not defined in the CIRCUIT section.
THIS EXTERNAL IS ALREADY CONNECTED TO NODE xx AT LINE nn	An external name has been used at two nodes.
THIS EXTERNAL CONNECTED TO NODE AT LINE nn, DEVICE PIN AT LINE nn	An external was referenced to two different nodes.
THIS INTERMEDIATE NAME HAS NOT BEEN DEFINED ANYWHERE	The intermediate name was not defined in the CIRCUIT section.
THIS IS A TESTER PIN NAME ASSIGNED TO NODE xx AT LINE nn	The adaptor pin is specified at two nodes.
THIS NAME HAS NOT BEEN DEFINED ANYWHERE	Device or node name not defined in CIRCUIT section.

THIS NAME WAS NOT FOUND

THIS NODE HAS BEEN GIVEN AN INTERMEDIATE NAME xx AT LINE nn

THIS PIN CONNECTED TO NODE AT LINE nn, DEVICE PIN AT LINE nn

THIS TESTER PIN PREVIOUSLY CONNECTED TO NODE xx AT LINE nn

B.2.4 Pass 2 Warnings

ADAPTOR PIN IS NOT CONNECTED TO ANY NODE

DEVICE PIN CONNECTED TO ITSELF

NO ADAPTOR PIN IS CONNECTED TO NODE xx AT LINE nn

NO DEVICES CONNECTED TO THIS NODE

THIS EXTERNAL ADDED TO NODE xx THOUGH NOT MENTIONED AT LINE nn

THIS PIN ADDED TO NODE xx THOUGH NOT MENTIONED AT LINE nn

B.2.5 Pass 3 Errors

IC NAME NOT FOUND

DIGITAL IC PIN HAS AN INVALID NAME, nn, IN COMPONENT LIBRARY

NO EXTERNAL IS CONNECTED TO NODE xx AT LINE nn

NODE xx HAS MORE THAN ONE EXTERNAL AT LINE nn

B.2.6 Pass 3 Warnings

IC TYPE xx IS CKTGEN TEST SKELETON
This message warns the user that digital tests may not have adequate inhibits because no inhibits exist in the test skeleton.

ONLY SPECIAL NAILS CONNECTED TO NODE xx AT LINE nn
Warns the user that no shorts test can be done at this node because no hybrid nail is connected.

NO ADAPTER PIN IS CONNECTED TO NODE xx AT LINE nn
This warning is repeated in pass 3 because of the probable importance of running shorts tests at all nodes.

nn ICs WITH FAST NAIL DEFICIT
There are not enough fast nails available for SpeedPlus testing on all ICs that have only fast tests in the libraries.

nn ICs OF TYPE CKTGEN TEST SKELETON
Number of ICs that have tests inserted via the CKTGEN test skeleton.

TOO MANY NAILS OF TYPE xx
There are not enough nails of the indicated type available on the system for nail assignment to satisfy all testing requrements.

nn H SCANNER BOARDS PLUS xx F SCANNER BOARDS EXCEEDS SYSTEM LIMIITS
The number of scanner boards necessary for nail assignment to assign the required H and F pins is greater than those available. Nail assignment will try to assign other types to solve the deficiency.

B.3 TEST PROGRAM TRANSLATOR ERRORS

B.3.1 Pass 1 and 2 Errors

BAD INTERNAL MSG	An error has occurred but an error in the translator has caused the error handler to become confused. This message should never appear in normal operation.
CAN'T CLOSE OBJ FILE	Either the symbol table (.SMT) file or the pass 2 temporary (TM2) file could not be closed.
DATA MANAGEMENT ERROR	
MAXIMUM ERROR COUNT EXCEEDED - TRANSLATION ABORTED	
NO MORE ROOM ON DISK	An error was encountered in writing an output file. While all problems of this nature are treated as a full disk, this message may also be caused by attempting to use a locked file.
SUBROUTINE ARGUMENTS NOT ALLOWED	

B.3.2 Pass 1 (Syntax) Errors.

MULTIPLY DECLARED VARIABLE	A variable has been declared more than once within the main program or a subroutine.
ONLY ONE DECLARATION ALLOWED	
UNDECLARED VARIABLE	A referance has been made to a variable which has not been declared.
VARIABLE IS OF WRONG TYPE	The type of the variable which has been referenced is inappropriate for the operation specified.
MISSING 'END' OR UNRECOGNIZED 'END' DUE TO OTHER ERR	The end of the input file stream was encountered without recognizing the 'END' keyword. This is usually due to an error somewhere in the main program confusing the translator to the extent that when the 'END' keyword is read, it is not recognized as such.
INVALID CHARACTER	A character has been encountered that is not recognized by the translator. The legal character set is the following: Characters A - Z: ABCDEFGHIJKLMNOPQRSTUVWXYZ Numerals 0 - 9: 0123456789 Operators: + - * / = > < Special symbols: ( ) [ ] : ; $ % _ ! '
LINE TOO LONG	A line which exceeded 80 characters in length was encountered. It must be broken up into two or more lines.
STRING TOO LONG	A string larger than 80 characters has been encountered. To create strings of length greater than 80 characters, concatenate two smaller strings.
INTEGER OUT OF RANGE	An integer constant with a magnitude larger than 32767 has been encountered.
MISSING */ FOR PREVIOUS COMMENT	A /* has been encountered without a matching */ to close the comment.
INVALID CONSTRUCT	
MISSING %	An I/O escape sequence begun by a % was not terminated by a matching %.
SSTK OVERFLOW	
NAME TOO LONG	An identifier with a length greater than 11 characters was found.
UNDECLARED SUBROUTINE	A reference has been made to an undeclared subroutine.

INCORRECT SUBROUTINE NAME	
MULTIPLY DEFINED FAIL BSTRING	The BSTRING "FAIL" may be only declared once.
CAN'T CLOSE TPG FILE	An error was encountered in attempting to close the input .TPG file. This is not a normal occurence, and is an indication of a system problem.
SYNTAX PASS OPEN ERROR	
CAN'T OPEN OBJ FILE	
BAD MON CNTRL REC	
SYNTAX ERROR	
FAIL VECTOR MAY NOT BE DECLARED AS GLOBAL	
NUMBER INCORRECTLY SPECIFIED	
RESERVED SYSTEM FILE NAME	
MULTIPLY DEFINED LABEL	A label has been defined in two different places
SEMICOLON IS REDUNDANT	An extra semicolon was found. While usually not serious, extra semicolons may cause a problem.
BEGIN RECOMPILE LABEL DOESN'T EXIST	The label specified as the begin label of a CHANGE command does not exist in the original test program. As the RECOMPILE statement is produced by the CHANGE "extractor" directly from the original test program, this message indicates tampering with either the .INC file, the RECOMPILE statement or both.
END RECOMPILE LABEL DOESN'T EXIST	The label specified as the end label of a CHANGE command does not exist in the original test program. As the RECOMPILE statement is produced by the CHANGE "extractor" directly from the original test program, this message indicates tampering with either the .INC file, the RECOMPILE statement or both.
STRING LACKS TERMINATING QUOTE	A string was found which did not have a terminating quote.
STATEMENT LACKS TERMINATING SEMICOLON	
.OBC FILE DOES NOT MATCH .SMT FILE	During a CHANGE command, the .SMT file was found to have been produced by a different TRANSLATE or CHANGE which produced the .OBC file. The information in a .SMT file is specific to the .OBC file which was created with it, so both files must be CHANGEd together.
PARAMETER IS MISSING	A subroutine parameter which was specified in a previous subroutine declaration or definition is missing from the declaration or definition being processed.

EXTRA PARAMETER ENCOUNTERED

A subroutine parameter was specified in the declaration or definition being processed, which was not specified in a previous declaration or definition of that subroutine.

A PARAMETER HAS BEEN SPECIFIED TWICE

PARAMETER [TYPE,LENGTH, and/or POSITION] DO(ES) NOT MATCH PREVIOUS USAGE

A subroutine parameter was specified in the declaration or definition being processed, with a type, length, and/or position different than that which was specified in a previous declaration or definition of that subroutine.

SUBROUTINE DECLARATION DOES NOT MATCH PREVIOUS USAGE

TARGET SYSTEM CONFLICTS WITH PREVIOUS TRANSLATION

ORIGINAL TRANSLATION MADE WITH OBSOLETE TRANSLATOR

STATEMENT MAY OVERFLOW RUN TIME STACK

B.3.3 Pass 2 (Object code generation) Errors

EXPRESSION TOO COMPLEX

The indicated expression is too complex for the translator to handle. Large expressions should be broken up, this helps with readability as well.

CAN'T OPEN LIS FILE
CAN'T OPEN TPG FILE
UNIT NUMBER NOT ALLOWED
PARAM LIST SEARCH ERROR
DRIVER NAME SEARCH ERROR
BAD DRIVER FUNCTION DATA
WRONG PARAMETER IN STMT
BAD MUX PIN LIST
BAD SCANNER PIN LIST
MISSING OR INCORRECT PARAMETERS IN STMT
AT CONSTRUCT NOT ALLOWED
ONLY ONE PIN LIST ALLOWED
TOO MANY PIN LISTS
DON'T SPECIFY CHANNEL FOR SHORTS
BAD MUX RAIL NAME
BAD PIN NUMBER
PIN LIST TOO COMPLEX, USE TWO STMTS
SYNTAX ERROR
DATA MANAGEMENT ERROR
CSTK UNDERFLOW
STMT TOO LONG, USE TWO STMTS
BAD PARSE CALL
BAD NAIL NUMBER
ONLY 2 ANALOG OR DIGITAL SIGNALS ALLOWED PER PAIR OF PINS

FAIL BSTRING NOT DECLARED OR TOO LARGE

An instrumentation statement was used without declaring a FAIL BSTRING, or the given declaration was too large.

FAIL BSTRING MUST BE GREATER THAN ZERO

The length of strings must be positive.

ONLY 20 SUBROUTINE PARAMETERS ALLOWED

An attempt was made to use an excessive number of parameters to a subroutine.

ONLY ONE CONNECTION MODE PER STMT
BAD DRIVER COMMAND DATA
DIGITAL STMT OUT OF SEQUENCE
DIGITAL LOOP NESTED
DIGITAL LACKS ASSIGN STMT
DIGITAL PIN SPEC ILLEGAL
EXCEPTION BRANCH NOT ALLOWED
LOOP MODIFIER INCORRECTLY USED
REDUNDANT PIN SPEC IN GROUP
TOO MANY PINS IN GROUP
GROUP OPERATOR IS NOT UNARY
GROUP OPERATOR IS NOT BINARY
DIGITAL SUBROUTINE NOT DEFINED
DIGITAL SUBROUTINE MULTIPLY DEFINED
BEGIN RECOMPILE LABEL IN WRONG PLACE
DIGITAL STMT HAS TOO MANY LABELS
DIGITAL VOLTAGE OUT OF RANGE
DATA MANAGEMENT FATAL ERROR
DIGITAL ITT OR IST OUT OF RANGE
MAXTIME OUT OF RANGE
DIGITAL BURST IS TOO LARGE
DIGITAL LOOP COUNT OUT OF RANGE
LOG/DIAG CAN'T BE USED HERE
OVERFLOW OBJECT CODE SEGMENT
ERROR CLOSING LISTING FILE

NO EXECUTABLE STMT IN WHILE-DO LOOP (POSSIBLY EXTRA ';' AFTER DO)

A WHILE-DO loop was found which had no executable statements after it. This is a strong indication of a ';' directly following the DO keyword. While not an error this often produces an infinite loop.

CONSTANT TOO LARGE
VARIABLE SPACE EXCEEDED
ONLY ONE BRACKETED EXPRESSION ALLOWED
SYNTAX PASS CLOSE ERROR
INVALID CHANNEL DESIGNATOR
INVALID PIN TYPE SPECIFIER
BRANCH TO UNDEFINED LABEL
PINS INCOMPLETELY SPECIFIED
PIN MULTIPLY ASSIGNED
PIN NOT ASSIGNED
PIN NOT CONNECTED
CONNECTED PIN CANNOT BE USED AS OUTPUT
ONE OR MORE PARAMETERS ARE MISSING

B.4 RUN-TIME DIAGNOSTIC MAJOR ERRORS

ARG OUT OF RANGE	The hardware does not support the value of an argument specified by the program.

BAD ARGUMENTS	The values, types, or numbers of arguments is not consistent with the instruction receiving them.
BAD DATA-TYPE	Data type, size, or quantity is not consistent with instruction requirement.
BAD HARDWARE VERSION	Revision level of module does not correspond with the identification table.
BAD 2270 INSTRUCTION	A statement was encountered which had an illegal op code.
BRANCH TO NONEXISTENT LABEL	
CAN'T EXECUTE CURRENT STATEMENT BECAUSE PREVIOUS BURST LEFT STATE ACTIVE	
CLOSE FAILURE	An error occurred while trying to close a file.
DEVICE FULL	
DIGITAL POWER SUPPLY FAILURE	
EXECUTE CODE ERROR	
GENERAL PRINT ERROR	
INVALID PROBE NAIL NUMBER	
HARDWARE CAN'T DO	Tried to cause a source instrument to overrange while performing a test.
HARDWARE MALFUNCTION	Function was not performed within the timeout period.
HARDWARE MALFUNCTION SpeedPlus cannot enter slave mode	SpeedPlus controller is not operating properly. Call service.
HARDWARE MALFUNCTION SpeedPlus ctlr hung in NOT DONE state	SpeedPlus controller is not operating properly. Call service.
I/O FAILURE	Error occurred while trying to input or output a file.
LOCKED FILE	
MANUAL ABORT	
MISSING HARDWARE	
NO [ ] BOARD	
NO FAIL VECTOR	A fail bit was not contained in an instruction that requires it.
NO SPEEDPLUS HARDWARE FOUND or wrong digital controller	Tried to run a SpeedPlus test when hardware was not available or when SpeedPlus controller not installed.

NO SUCH FILE	
NO SYNCHRONIZATION ACHIEVED	The synchronization to an on board clock for a SpeedPlus test was not achieved so no burst was executed.
NOT IMPLEMENTED	The defined function is not implemented in this hardware.
.OBC WAS FILE TRANSLATED FOR 2270	
.OBC WAS FILE TRANSLATED FOR 2271	
.OBC WAS FILE TRANSLATED FOR 2272	
.OBC WAS FILE TRANSLATED FOR 2275	
OPEN FAILURE	An error occurred while trying to open a file.
PRIVATE	
Ref Voltage out of 936 D/S chip range	A logic voltage was specified in the program outside the range (-6 to +10 volts) of the new driver/sensor chip. The new chip is used in the driver/sensor boards for all SpeedPlus options, many standard digital sections in 2271/72 systems and all 2275 testers.
SPEEDPLUS TIMEOUT OCCURRED	SpeedPlus controller contnued to run after the digital controller finish and until timeout occurred.
SOFTWARE FATAL BUGGY	
STRING RANGE ERROR	
SYNCHRONIZATION LOST	The synchronization of test steps to the clock was lost during the burst. The test result may be incorrect.
TIMEOUT OCCURRED	The burst just performed exceeded the maximum time allowed for a burst by the digital controller. The most likely cause is a missing trigger.
TOO FEW ARGUMENTS	
TOO MANY ARGUMENTS	
TOO MANY FILES	The program tried to open more channels than the system can support at one time.
TOO MANY SHORTED ICS	It is unsafe to continue testing with this many shorts. Testing is aborted.
UNKNOWN FILE	The file name was not found.
2270 STACK OVER/UNDER-FLOW	The stack used to store subroutine calls, arguments, etc. ran out of space or had an unequal number of push and pops.
WRITING A READ-ONLY SEGMENT	

B.5 RSX-11M SYSTEM ERROR CODES

For a complete set of DEC I/O error codes refer to the IAS/RSX-11M I/O Operations Reference Manual available from Digital Equipment Corp.

Common RSX-11M System I/O Error Messages

Message	Code	Meaning
.DNR	3	Device not ready.
.VER	4	Parity error on device.
.DNA	7	Device not attached.
.EOF	10	End-of-file detected.
.WLK	12	Write attempted to locked unit.
.DAO	13	Data overrun.
.SRE	14	Send/Receive failure.
.ABO	15	Request terminated.
.PRI	16	Privilege violation.
.DFU	24	Device full.
.NSF	26	No such file.
.LCK	27	Locked from read/write access.
.WAC	29	Accessed for write.
.CKS	30	File header checksum failure.
.CLO	38	File was not properly closed.
.NBF	39	No buffer space available for file.
.RBG	40	Illegal record size.
.NBK	41	File exceeds space allocated, no blocks.
.FEX	49	Rename - a new file name already in use.
.BDR	50	Bad directory file.
.RNM	51	Cannot rename old file system.
.BDI	52	Bad directory syntax.
.FOP	53	File already open.
.BNM	54	Bad file name.
.BDV	55	Bad device name.
.BBE	56	Bad block on device.
.DUP	57	Enter - duplicate entry in directory.
.FHE	59	Fatal hardware error on device.
.NFI	60	File ID was not specified.
.ISQ	61	Illegal sequential operation.
.BVR	63	Bad version number
.BHD	64	Bad file header.
.OFL	65	Device offline.
.NNN	68	No such node.
.TMM	71	Too many outstanding messages.
.NDR	72	No dynamic space available.
.IES	82	Invalid escape sequence.
.PES	83	Partial escape sequence.
.ALC	84	Allocation failure.
.ULK	85	Unlock error.
.WCK	86	Write check failure.
.IDU	92	Invalid device or unit.

NOTE

In general, if the system reports a negative error number, it is repeating one of the RSX-11 I/O numbers above.

GenRad

300 Baker Avenue
Concord, Massachusetts 01742

USA
***Boston** • (Sales) Waltham, MA 02254
Tel: (617) 890-4900 • TWX: 710 324 0893
(Service) Concord, MA 01742
Tel: (617) 369-4400 • TWX: 710 347 1051
***Chicago** • 1156 Shure Drive, Arlington Heights, IL 60004
Tel: (312) 577-1881 • TWX: 910 291 1209
***Cincinnati** • Park 50, 2002 Ford Circle, Milford, OH 45150
Tel: (513) 831-9210 • TWX: 810 460 8323
***Dallas** • 1121 Rockingham Lane, Suite 100, Richardson, TX 75080
Tel: (214) 234-3357 • TWX: 910 867 4771
***Los Angeles** • P.O. Box 19500, 17361 Armstrong Avenue,
Irvine Industrial Complex, Irvine, CA 92714
Tel: (714) 540-9830 • TWX: 910 595 1762
***New York** • 22-08 Route 208, Fair Lawn, NJ 07410
Tel: (NJ) (201) 797-8001; (NY) (212) 964-2722 • TWX: 710 988 2205
***Orlando** • 3751 Maguire Blvd., Suite 170, Orlando, Florida 32803
Tel: (305) 894-4303 • TWX: 810 850 0270
***San Francisco** • 2855 Bowers Avenue, Santa Clara, CA 95051
Tel: (408) 727-4400 • TWX: 910 338 0291
***Washington, D.C.** • 1701 Research Boulevard, Rockville, MD 20850
Tel: (301) 424-6224 • TWX: 710 828 9783

CANADA
***Toronto** • 307 Evans Avenue, Toronto, Ontario M8Z 1K2
Tel: (416) 252-3395 • TLX: 06-967624
Montreal • Tel: (514) 747-1052 • TLX: 05-826652

ASIA
Japan • Tokyo Electron Limited, Shinjuku-Nomura Bldg.,
1-26-2 Nishi-Shinjuku, Shinjuku-Ku, Tokyo 160, Japan
Tel: 343-4411 • TLX: 232-2240 Labtel J
Other countries in Asia and Pacific • Tel: 344-9245 (Tokyo)

*Service Facilities Available

EUROPE
***England** • Norreys Drive, Maidenhead, Berkshire SL6 4BP
Tel: (0628) 39181 • TLX: 851-848321
Best • Tel: (0) 49984240 • TLX: 844-59535
***Milan** • Tel: (02) 8466541 • TLX: 843-320373
***Munich** • Tel: (089) 496001 • TLX: 841-529917
***Paris** • Tel: (01) 7970739 • TLX: 842-220991
Rennes • Tel: (099) 535154
Rome • Tel: (06) 455839
Wiesbaden • Tel: (06121) 370057 • TLX: 841-04186489
***Zurich** • Tel: (01) 552420 • TLX: 845-53638

LATIN AMERICA
Contact WALTHAM, MA 02254 USA
Cable GENRAD Waltham (MASS.)

Printed in USA